

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra Informatiky

Portál pro sdílení a archivaci faktur v důvěryhodné podobě
Portal for Sharing and Archiving Invoices in a Trusted Form

Zadání diplomové práce

Student:

Bc. Tomáš Raszyk

Studijní program:

N2647 Informační a komunikační technologie

Studijní obor:

2612T025 Informatika a výpočetní technika

Téma:

Portál pro sdílení a archivaci faktur v důvěryhodné podobě
Portal for Sharing and Archiving Invoices in a Trusted Form

Jazyk vypracování:

čeština

Zásady pro vypracování:

Cílem práce je vytvořit fakturační portál pro důvěryhodnou archivaci vydaných faktur v elektronické podobě. Samotné faktury budou vznikat v externích ERP (Enterprise Resource Planning) systémech (např. SAP). Fakturační systém umožní ukládání faktur, sdílení mezi více uživateli a přikládání naskenovaných příloh. Důležitou vlastností portálu bude ukládání faktur a jejich příloh v podobě, která důvěryhodně zajistí jejich neměnnost.

Systém umožní:

1. Import faktur z externích ERP systémů.
2. Skenování příloh označených čárovým kódem a jejich automatické propojení s odpovídajícím dokladem již uloženém ve fakturačním portálu.
3. Rozesílání informací o dostupnosti nové faktury uživatelům fakturačního portálu.
4. Vytvoření jediného dokumentu, který obsahuje fakturu a všechny její přílohy.

Práce bude obsahovat:

1. Popis technologií používaných pro komunikaci mezi ERP systémy.
2. Popis technologií využívaných pro zajištění důvěryhodnosti dokumentů.
3. Analýzu požadavků na fakturační portál
4. Analýzu a návrh výše popsaného fakturačního portálu.
5. Implementaci a popis řešení fakturačního portálu.

Seznam doporučené odborné literatury:

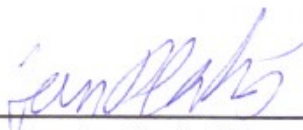
- [1] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides (Gang of Four): Návrh programů pomocí vzorů. Grada. Praha 2003. ISBN 8024703025
- [2] CHUVAKIN, Anton, Kevin J. SCHMIDT, Chris PHILLIPS a Patricia MOULDER. Logging and log management: the authoritative guide to understanding the concepts surrounding logging and log management. Amsterdam: Elsevier/Syngress, [2013]. ISBN 978-1597496353.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. David Ježek, Ph.D.**

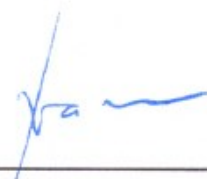
Datum zadání: 01.09.2019

Datum odevzdání: 30.04.2020



doc. Ing. Jan Platoš, Ph.D.
vedoucí katedry





prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlášení Studenta

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Jirkově 9. května 2020



Poděkování

Na tomto místě bych rád poděkoval všem, kteří mi poskytli svou pomoc a podporu při vytváření této práce. Hlavně vedoucímu práce panu Ing. Davidu Ježkovi za cenné rady, vedení a konzultace, které mi poskytoval během tvorby této práce. Ale také za ohromnou trpělivost, kterou se mnou při psaní této práce měl. Bez jejich pomoci by tato práce nikdy nevznikla.

Abstrakt

Cílem diplomové práce je prostudovat problematiku elektronické fakturace, jakožto moderního prostředku, kterým lze jednoduše zefektivnit příjem a odesílání faktur mezi ekonomickými subjekty. Dle zjištěných požadavků, které jsou kladeny na elektronickou fakturaci nejen ze strany zákonů České republiky, ale také Evropské unie, pak navrhnout takové řešení, které vyhoví všem legislativním požadavkům a zároveň umožní k takovéto faktuře připojit libovolnou přílohu. Systém následně takto legislativně platný kompletní dokument bude schopen distribuovat mezi zákazníky portálu a samotné doklady bude archivovat k dalšímu zpracování.

Klíčová slova: Elektronická faktura, XML, kompletace, HASH, PDF, certifikát, časové razítko, čárový kód, metadata, BPMN, EFZ

Abstract

The aim of the diploma thesis is to study the issue of electronic invoicing, as a modern means by which it is easy to streamline the receipt and sending of invoices between economic entities. According to the identified requirements, which are placed on electronic invoicing not only by the laws of the Czech Republic, but also the European Union, then design a solution that meets all legislative requirements and also allows you to attach any attachment to such an invoice. The system will then be able to distribute the complete document legally valid in this way among the portal's customers and will archive the documents themselves for further processing.

Keywords: Electronic invoice, XML, completion, HASH, PDF, certificate, time stamp, barcode, metadata, BPMN, EFZ

Seznam použitých zkratek a symbolů

ERP	- Plánování podnikových zdrojů nebo někdy také podnikový informační systém
SAP	- Celosvětově známý ERP produkt pokrývající všechny oblasti řízení podniku (finance, odbyt, majetek, sklady atd.)
IDoc	- Intermediate document je nativní formát informačního systému SAP, který lze využít pro export a import obchodních a logistických dokladů
XML	- Formát dokumentu s libovolně vnořenými elementy, které mohou obsahovat další atributy
EDI	- Elektronická výměna dat je výměna strukturovaných zpráv mezi počítači, respektive mezi počítačovými aplikacemi
SSO	- Single Sign-On umožní použít jedno přihlášení do více aplikací zároveň
HASH	- Hašovací funkce je matematická funkce pro převod vstupních dat do malého čísla. Výstup hašovací funkce se označuje výtah, miniatura, otisk, fingerprint či hash
PDF	- Portable Document Format je souborový formát pro ukládání dokumentů nezávisle na softwaru i hardwaru, na kterém byly pořízeny. Soubor typu PDF může obsahovat text i obrázky, přičemž tento formát zajišťuje, že se libovolný dokument na všech zařízeních zobrazí stejně
Metadata	- Metadata představují dodatečný soubor informací, který je vytvořen a uložen pro každý dokument
BPMN	- Business Process Model and Notation je grafická notace sloužící k modelování podnikových procesů
EFZ	- Elektronická fakturace zákazníkům
DMS	- počítačový systém určený ke správě elektronických dokumentů a zdigitalizovaných papírových dokumentů s pokročilou možností vkládání metadat nebo tagů
DI	- Vkládání závislostí (Dependency injection) je v objektově orientovaném programování technika pro vkládání závislostí mezi jednotlivými komponentami programu tak, aby jedna komponenta mohla používat druhou, aniž by na ni měla v době sestavování programu referenci.
Framework	- Framework je softwarová struktura, která slouží jako podpora při programování a vývoji a organizaci jiných softwarových projektů.

Obsah

1 ÚVOD.....	5
2 CÍL PRÁCE	6
2.1 Důvěryhodný archiv	6
2.2 Kompletace faktur	7
2.3 Distribuce faktur	7
2.4 Rozsah projektu	8
2.5 Pokračování vývoje archivu	8
3 LEGISLATIVA ELEKTRONICKÉ FAKTURACE	9
3.1 Elektronická časová razítka	10
3.2 Zabezpečení dat	10
3.2.1 Skartace dat.....	12
3.3 Věrohodnost dat	12
3.3.1 Prokázání důvěryhodnosti dat.....	13
4 ANALÝZA POŽADAVKŮ	14
4.1 Business procesy	14
4.2 Příprava faktur	16
4.3 Příprava příloh	17
4.4 Kompletace faktur	19
5 NÁVRH SYSTÉMU	20
5.1 Aplikační rozhraní	20
5.2 XML komunikace.....	21
5.3 Metadata a webové služby	23
5.4 Technický koncept.....	24
6 IMPLEMENTACE SYSTÉMU	25

6.1 Požité technologie.....	25
6.2 Databázový model	27
6.3 XML modely.....	28
6.4 DinkToPdf	29
6.5 iTextSharp.....	30
6.6 Digitální podpis	31
6.7 Free Spire.Barcode	33
6.8 PDF Sharp.....	35
6.9 SendGrid	35
6.10 Diagram - JobSchedulerBIPProvider	36
7 FUNKCE ARCHIVU	38
7.1 Uživatelské rozhraní.....	39
7.2 Administrace EFZ.....	39
7.2.1 Číselníky	41
7.2.2 Import dat	41
7.2.3 Definice systémových úloh.....	42
7.2.4 Logy.....	43
8 VYHODNOCENÍ PROJEKTU	44
ZÁVĚR	47
LITERATURA	48
PŘÍLOHY.....	49

Seznam obrázků

Obrázek 1: Zajištění integrity dat v archivu	11
Obrázek 2: Obecný pohled na fungování systému.....	15
Obrázek 3: BPMN proces vystavení faktury	16
Obrázek 4: BPMN scénář č. 1 pro skenování přílohy	17
Obrázek 5: BPMN scénář č. 2 pro skenování přílohy	18
Obrázek 6: BPMN scénář č. 3 pro skenování přílohy	18
Obrázek 7: BPMN proces kompletace faktury	19
Obrázek 8: Struktura a povinné segmenty pro faktury	21
Obrázek 9: Struktura a povinné segmenty pro objednávky	22
Obrázek 10: Struktura a povinné segmenty pro dodávky	22
Obrázek 11: Technický koncept archivu EFZ	24
Obrázek 12: Znázornění logiky portálu	26
Obrázek 13: Databázový model archivu EFZ	28
Obrázek 14: Aplikace XCA pro generování certifikátu.....	33
Obrázek 15: Nekompletní diagram Časovače úloh.....	37
Obrázek 16: Základní diagram užití systému	38
Obrázek 17: Domovská stránka archivu EFZ.....	38
Obrázek 18: Text informačního e-mailu o nově vystavené faktuře	39
Obrázek 19: Kompletní sekce archivu EFZ	39
Obrázek 20: Formulář pro ruční nahrání přílohy	40
Obrázek 21: Diagram s přehledem funkcí dostupných pro administrátora	40
Obrázek 22: XML vstupy do archivu EFZ.....	41
Obrázek 23: Přehled systémových úloh	42
Obrázek 24: Stav běhu spuštěných úloh	43
Obrázek 25: Výpis chyby neúspěšného importu naskenované přílohy	43
Obrázek 26: Úspěšně zpracování přílohy s více čárovými kódy	45
Obrázek 27: Problém s ověřením certifikátu	45

Seznam kódů

Kód 1: Ukázka vytvoření DI	25
Kód 2: Ukázka načítání dat z DB kontextu	27
Kód 3: Ukázka definice používaných XML modelů	29
Kód 4: Ukázka importu XML dat pro vytvoření faktury	29
Kód 5: Ukázka vytvoření PDF faktury	30
Kód 6: Ukázka kompletace PDF dokumentu	31
Kód 7: Příznaky na faktuře pro kompletaci	31
Kód 8: Ukázka načtení (EFZ.pfx) podpisového certifikátu	32
Kód 9: přidání časového razítka	32
Kód 10: Ukázka čtení čárových kódů z obrázku	34
Kód 11: Ukázka prohledávání DB dle vyčtených záznamů	34
Kód 12: Ukázka úspěchu či neúspěchu vytěžení dat	35
Kód 13: Ukázka konverze obrázku do PDF	35
Kód 14: Ukázka rozesílání e-mailových notifikací	36

1 ÚVOD

Tato práce se bude věnovat tématu elektronické fakturace a požadavkům, které jsou na tento segment kladeny nejen ze strany platných zákonů České republiky. Jelikož je mnoho nejasností kolem elektronické fakturace obecně (platnost takových dokumentů, jak je archivovat, důvěryhodnost, apod.) objasníme si některé pojmy a oblasti, které jsou s elektronickou fakturací úzce spojeny a bez nichž bychom nebyly schopni splnit požadavky na systém dle platné legislativy.

Vysvětlíme si některé oblasti, bez jejichž pochopení by nebylo možné správně a v souladu s legislativou splnit požadavky na systém. Je nutné pochopit, že odeslání faktury pomocí e-mailové pošty rozhodně nesplňuje požadavky na elektronickou fakturaci a co je tedy nutné splnit, aby se elektronická faktura brala jako platný doklad. K tomu je potřeba vyhovět technickým aspektům kladeným na informační systémy obecně.

Ještě než přistoupíme k samotnému návrhu aplikace pro distribuci a archivaci faktur, seznámíme se s jednotlivými možnostmi, jakým způsobem mohou mezi sebou systémy komunikovat a vyměňovat si data. Tato analýza bude sloužit k technickému návrhu aplikace, která bude jako jeden ze vstupu mít rovněž zahrnuty business požadavky na samotný systém. Bez těchto požadavků by vývoj řešení nebyl příliš velkým přínosem a jeho použitelnost by mohla být minimální.

Implementační část již představí samotné řešení aplikace pro distribuci faktur, vysvětlí vnitřní logiku systému, dle níž fungují veškeré procesy naplňující cíle práce. Podíváme se na systém ze dvou různých pohledů, jak z pohledu běžného uživatele (tedy zákazníka), tak z pozice administrátora, jež definuje celou logiku systému a nastavuje jednotlivé procesy.

Jelikož práce nastiňuje problematiku elektronické fakturace jen okrajově, tak si v závěru vyhodnotíme, jak a zda se podařilo splnit všechny cíle a vzhledem k hotovému řešení si nastíníme i další možnosti, kterým by se aplikace mohla ubírat a rozvíjet tak svůj potenciál do budoucna.

2 CÍL PRÁCE

Cílem práce je vytvoření dlouhodobého důvěryhodného archivu pro distribuci a archivaci elektronických faktur. Podobných řešení dnes existuje mnoho a místo srovnání jednotlivých řešení se v práci zaměřím na požadavky, kladený nejen ze strany uživatelů (respektive zákazníků), ale také ze strany platných zákonů České republiky. Pokud budeme mít zanalyzovány dobře a detailně všechny vstupy, lze pak jednodušeji navrhnout cílové řešení, které bude vyhovovat všem požadavkům kladeným na systém přicházející z vně systému.

Navrhnuté řešení bude navrženo a implementováno velice přehledně a pokud možno s uživatelsky přívětivým rozhraním, ať je práce intuitivní, jednoduchá a v žádném případě neodrazuje od používání finálního řešení. Proto bude práce vycházet z detailní analýzy, která zohledňuje zmíněné aspekty a je navržena v jejich souladu. Než se však dostanu k samotnému řešení, nastíním pokud možno detailně a přesně jednotlivé pojmy a požadavky, které jsou s elektronickou fakturací úzce spojeny.

2.1 Důvěryhodný archiv

Pod pojmem archiv si můžeme představit budovu, místnost, popř. i obyčejnou skříň, jež slouží k uchovávání, evidování, ochraně a zpřístupnění např. cenných písemných dokumentů. Aby byl však archiv zároveň důvěryhodný, musí splňovat určité parametry a vlastnosti. Takový archiv umožňuje dlouhodobě a důvěryhodně archivovat dokumenty. Protože cílem práce je navrhnout portálové řešení, musí archiv zároveň garantovat integritu vložených dat, stejně tak prokázat původ archivovaných elektronických dokumentů v čase. Toho lze docílit použitím časových razítek poskytovaných nezávislou certifikační autoritou.

Řešení musí být v souladu se všemi legislativními požadavky a předpisy a zároveň se řídit doporučením na úrovni Evropské unie. Takto koncipované důvěryhodné archivy se používají zejména pro archivaci daňových dokladů, smluv, digitalizovaných dokumentů apod. Lze ho následně integrovat s ostatními IT systémy v prostředí firem nebo státní správy. Obsah archivu je přístupný pomocí jednoduchých rozhraní a lze s ním pracovat v různých režimech. Taktéž se dá integrovat s dalšími systémy a může sloužit jako samostatná pojistka vedle funkčních DMS ¹ systému.

Archiv má být navržený takovým způsobem, aby nebyl závislý na konkrétním datovém formátu archivovaných dokumentů, nebo jejich typu. Měl by být datově nezávislý. Do archivu lze nahrávat mimo automatických procesů další dokumenty manuálně a k již archivovaným dokumentům by mohl umožnit doplnit metadata.

¹ DMS - počítačový systém určený ke správě elektronických dokumentů a zdigitalizovaných papírových dokumentů s pokročilou možností vkládání metadat nebo tagů.

2.2 Kompletace faktur

Faktury jednou zaslané do archivu musí být archivovány, ale zároveň musí mít nějakou přidanou hodnotu, aby se naplnil jednak cíl této práce, ale zároveň aby nabídl něco navíc. Tím benefitem bude doplnění faktur o jejich přílohy, které jsou relevantní vzhledem ke svému obsahu k jednotlivým fakturám. Tyto přílohy musí být stejně jako faktury v elektronické podobě, ale jejich původ, tedy metoda vytvoření dané přílohy není závislá na žádném konkrétním řešení. Může se tak jednat o další elektronická data získaná integrací na jiná IT řešení (viz následující kapitoly), může se však jednat o dokument získaný ručním skenováním na multifunkčních zařízeních, použitím mobilního zařízení a nafocení daného dokumentu apod. Možnost ručního nahrání příloh bude vysvětlena později.

Nahrána příloha bude uložena dle domluvených podmínek v archivačním modulu portálu a během uložení dojde k vytěžení metadat potřebných pro proces kompletace a indexace. Faktury budou následně automaticky (lze však nastavit ruční kompletaci) propojeny s nově uloženými přílohami. Takto zkompletovaný dokument bude opatřen příznakem, že jsou kompletní a dojde k odstranění jednotlivých dokumentů. Označená faktura bude předána portálu ke zpracování, kdy dojde k vytvoření jediného PDF dokumentu. Tento dokument se včetně všech příloh podepíše elektronickým podpisem a uloží v důvěryhodném archivu.

2.3 Distribuce faktur

Obecně je vytvořená faktura v libovolném IS systému připravena k distribuci k zákazníkům standardní formou, což znamená tisk faktury na papír a odeslání faktury touto cestou. Někdy se také samotná faktura posílá jako příloha e-mailové pošty, což není vhodné z mnoha důvodů (viz kapitola o legislativě).

V portálu však zvolím jiný druh distribuce faktur a to zaslání pouze notifikace, o nově vytvořeném dokumentu. Příjemce faktury obdrží e-mailem kromě textu, ve kterém je o novém dokladu informován, také odkaz na fakturační portál, kde si fakturu společně s přílohami vyzvedne z webového prostředí. Tato aktivita zároveň může být v případě potřeby evidována pro potřeby potvrzení o vyzvednutí faktury.

Příjemci faktur však nebudou k fakturám přistupovat pouze pomocí rozesílaných notifikací, ale budou mít přístup na portál, kde uvidí všechny své dokumenty. Přístupy budou jedinečné pro každého zákazníka, každý uvidí pouze doklady náležící konkrétnímu zákazníkovi.

Jelikož se jedná o dlouhodobý důvěryhodný archiv, který nemusí být z mnoha důvodů přístupný pro všechny zákazníky (již jsem v úvodních kapitolách nastínil, že takový archiv lze používat nejen pro fakturaci, ale i o jiné typy dokumentů), dá se využívat také pouze jako archiv pro vygenerované PDF faktury (ve smyslu faktur, které se mohou posílat stále v papírové podobě), jež se nemusí následně archivovat ve fyzické podobě pro potřeby budoucích kontrol ze strany kontrolních orgánů

2.4 Rozsah projektu

Stejně jako u libovolného IT projektu (lze uplatnit i mimo IT odvětví), je důležité si na úvod stanovit rozumný scope (tedy rozsah) celého projektu, ideálně na základě aktuálního stavu situace (as-is) a stanovit si požadovaný cíl (to-be). Pokud bych z počátku špatně podchytil analýzu projektu a nedokázal správně pokritizovat jednotlivé požadavky na systém, mohlo by se celé řešení zhroutit.

Prioritou a vlastně i celým smyslem projektu je vytvořit takové řešení, která umožní kompletovat faktury s jejich přílohami a zároveň takto vytvořený jediný dokument dokáže opatřit elektronickým podpisem a časovým razítkem. Tento dokument následně nebude distribuovat ve formě přílohy e-mailové pošty, ale bude pouze dostupný v archivu zákazníkům a zároveň bude dlouhodobě a důvěryhodně všechny dokumenty archivovat.

Samotné faktury se nebudou pořizovat v archivu, ale budou importovány z externích systému vhodnou metodou. Nedílnou součástí projektu je také skenování příloh, které budou opatřeny čárovým kódem. Data vytěžená z těchto čárových kódů budou použita pro indexaci a správné párování s fakturami dle dohodnutých parametrů.

Posledním cílem je distribuce kompletních, podepsaných a důvěryhodných dokumentů zákazníkům. Tahle funkce není kritická nebo jinak důležitá pro chod archivu, slouží však pro zefektivnění celého procesu.

2.5 Pokračování vývoje archivu

Nyní když jsem uvedl rozsah celého projektu, obecně nastíním možný budoucí vývoj a rozšíření, kam by se archiv mohl ubírat. Pokud by se archiv aktivně využíval a byl nasazen do produkčního prostředí, samotná hodnota archivu by k aktivnímu používání pravděpodobně byla málo. Bylo by potřeba dalších funkcí a doplňků, aby to mělo smysl.

Směr, kterým by se dal archiv rozšířit je mnoho (např. sledování stavu faktur, potvrzení o převzetí, systém reportů, statistik atd.) a jakmile představím samotné řešení (kapitoly o implementaci a používání archivu), nastíním i jednotlivé směry vývoje, které jsem nebyl schopný do aktuální podoby archivu přidat.

To jsou však technické aspekty řešení, ke kterým je potřeba znát nejen požadavky uživatelů systému, ale také legislativu prostředí, ve kterém bude systém používán. Legislativou se bude zabývat následující kapitola.

3 LEGISLATIVA ELEKTRONICKÉ FAKTURACE

Legislatura je pojem, který má čtyři významy: Zákonodárná moc, právní řád daného státu, legislativní proces a zákonodárský sbor. Legislatura moderního demokratického právního státu (ve všech čtyřech významech) vychází a je založena na koncepci veřejné moci [1]. Pro účely této práce vysvětlím co je to elektronická fakturace z pohledu zákona.

Elektronická faktura (dále jen „e-faktura“) je dokumentem v digitální podobě (elektronickým dokumentem) podle ustanovení § 2 zákona č. 499/2004 Sb., o archivnictví a spisové službě a o změně některých zákonů a nařízení eIDAS. Může být doručena ve strukturovaném nebo nestrukturovaném datovém formátu. Může být účetním záznamem podle zákona č. 563/1991 Sb., o účetnictví, daňovým dokladem podle zákona č. 235/2004 Sb., o dani z přidané hodnoty, může sloužit i pro jiné účely (např. jako dodací list, záruční list, důkazní prostředek apod.). [2]

Tohle je základní definice elektronické faktury, která musí být dodržena v průběhu implementace a realizace elektronické fakturace do organizačních jednotek. Specifika a další definice (např. rozbor Evropské směrnice 2014/55/EU) si zde rozebírat nebudeme, protože to není cílem této práce. Uvedu však definované technické standardy, které se musí v souladu se směrnicí 2014/55/EU dodržet:

- Evropská norma pro elektronickou fakturaci EN 16931-1:2017 ²
- Formáty / syntaxe dle Evropské směrnice 2014/55/EU čl.3, odst. 2:
 - a) formát / syntax UBL 2.1 ISO/IEC 19845:2015
 - b) formát / syntax UN/CEFACT CII (Cross Industry Invoice)
- Formát ISDOC/ISDOCX (Information System Document) verze 5.2 a vyšší (dle Usnesení vlády č. 347/2017 a vyhlášky č.194/2009 Sb).

Jakákoliv odchylka od tohoto pravidla by představovala porušení směrnice. Za zmínku však stojí, že směrnice 2014/55/EU nepředepisuje, která syntaxe by měla být použita pro elektronickou fakturaci při zadávání veřejných zakázek. Pouze uvádí, které syntaxe jsou veřejní zadavatelé povinni akceptovat. Z čehož vyplývá, že ostatní syntaxe neuvedená ve standardech výše se budou používat i nadále, včetně přeshraniční výměny. To je případ českého formátu ISDOC, který není součástí uvedené směrnice, ale přesto se používá nadále.

Na základě výše popsaných informací je zřejmé, jaké technické parametry musí faktury splňovat, aby vyhověly legislativě. Zároveň také platí, že faktury poslané prostřednictvím e-mailové pošty, které nejsou elektronicky podepsané, nebo neobsahují časové razítko, tak nesplňují legislativní požadavky na neměnitelnost dokumentu. Tento požadavek docílíme jinými technickými prostředky, které si probereme podrobněji. Pochopení jejich funkce je klíčové pro naplnění cílů této práce.

Poznámka k Evropské normě EN16931 – pomocí online dostupného validátoru, což je webová služba, lze provést sémantickou a syntaktickou validaci (kontrolu) se schváleným formátem elektronické faktury dle normy EN 16931.

² Norma stanoví sémantický datový model základních prvků elektronické faktury. Sémantický model obsahuje jen nezbytné informační prvky, které musí elektronická faktura obsahovat, aby byl zajištěn soulad s právními (i daňovými) předpisy a aby byla umožněna požadovaná interoperabilita.

3.1 Elektronická časová razítka

Časové razítko zajišťuje přiřazení aktuálního časového údaje k existujícím datům, informacím, souborům nebo událostem. Připojuje se tedy k dokumentům, datům či elektronickému podpisu, u kterých chceme mít v budoucnu možnost ověřit, že v této podobě existovaly před časem uvedeným v časovém razítku. [3]

Vydávání kvalifikovaných elektronických časových razítek je upraveno nařízením EU č. 910/2014 o elektronické identifikaci a službách vytvářejících důvěru pro elektronické transakce na vnitřním trhu (eIDAS) a zákonem č.297/2016 Sb., o službách vytvářejících důvěru pro elektronické transakce. K jejich vydávání je oprávněn pouze kvalifikovaný poskytovatel služeb vytvářejících důvěru. Časové razítko vydané nezávislou časovou autoritou dokazuje existenci dokumentu v konkrétním čase. Mám-li k dokumentu připojené takové časové razítko, je to neměnitelný důkaz, potvrzující existenci dokumentu v přesně takové podobě. Tohle je nutné pro účely ověřování.

Funkce otisků časových razítek se dá aplikovat taktéž na podpisové certifikáty, čímž opět ověřím platnost daného elektronického podpisu v čase. To je další úroveň kontroly během prokazování neměnitelnosti dokladů v čase.

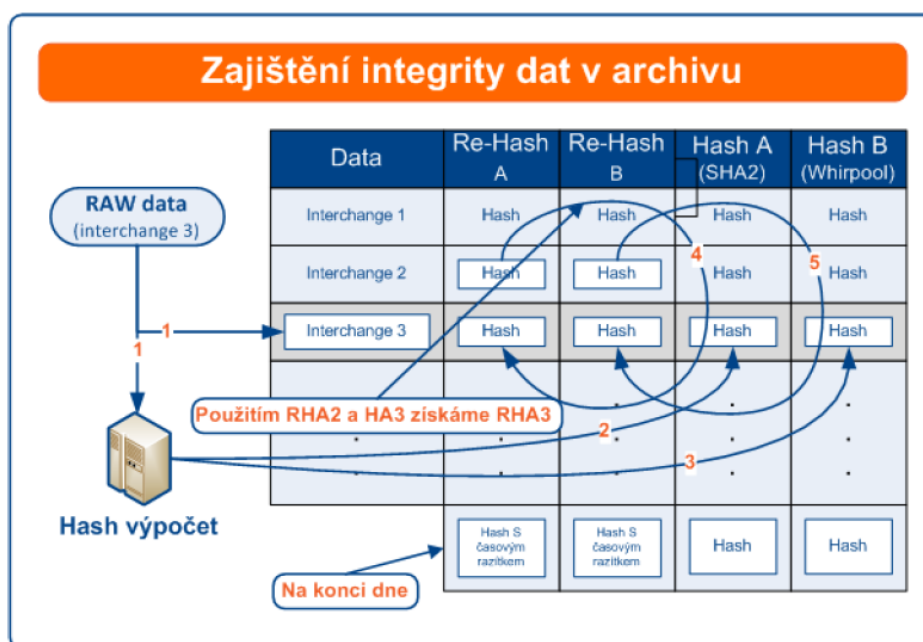
Samotná funkce časového razítka bude popsána v pozdější kapitole, nyní jen pro představu funkčnosti uvedu, že archiv (respektive aplikační server) komunikuje prostřednictvím zabezpečeného SSL požadavku se serverem časové autority. Struktura síťové komunikace mezi archivem a servery časové autority nebude pokryta touto prací.

3.2 Zabezpečení dat

Uvedl jsem, že pro dodržení důvěryhodnosti archivu je nutné dodržet a garantovat integritu vložených dat a zároveň prokázat původ archivovaných dokumentů v čase, k čemuž slouží funkce časových razítek. Na integritu však časové razítko nestačí a je potřeba zavést jinou technologii. Ta je svým rozsahem bohužel příliš komplexní a není v možnostech této práce, aby problematiku kompletně obsáhla do svého rozsahu. Lze však stručně definovat a popsat, jak lze data zabezpečit a dodržet podmínky na integritu dat.

Integrita dat je zajištěna dvěma (lze však definovat libovolný počet řetízků) nezávislými řetězci otisků dokumentů využívajícími různé algoritmy k jejich vytvoření s časovými razítky od autorizovaného poskytovatele. Tato technika zaručuje integritu i u dokumentů, které nejsou digitálně podepsány. Zajištění původu v čase poskytují řetězce otisků a časová razítka. Celý proces pracuje dle následujících pravidel (obrázek č.1):

- Otisk (HASH) se vypočítá z textu ukládaného dokumentu (1). Archiv používá dva různé algoritmy a ukládá výsledky do sloupců HASH1 (2) a HASH2 (3) databázové tabulky.
- Otisky (HASH) jsou zřetěžené. REHASH1 se vypočítá jako $REHASH1 = HASH1 + REHASH1$ předchozího ukládaného dokumentu (4). REHASH2 se vypočítá stejným způsobem (5). Smazání řádku nebo vytvoření nového otisku v databázové tabulce by vedlo k přerušení řetězce.
- Na konci dne jsou všechny řetězce opatřeny časovým razítkem autorizovaného poskytovatele REHASH1 a REHASH2. Časové razítko, REHASH1 a REHASH2 jsou uloženy v databázové tabulce pro budoucí možnost ověřování nebo kontroly.



Obrázek 1: Zajištění integrity dat v archivu

Proces archivace dle nastavení uloží dokument, který se stává součástí řetízku otisků dat (Hash Chain), čímž je zabezpečena jeho nezměnitelnost. Na konci dne pak proběhne automatická úloha, která přidá časové razítko na řetízek otisků, které zajistí jednoznačné prokázání původu archivovaných dat v čase.

Alternativou k razítkování řetízků otisků je razítkování na úrovni jednotlivých dokumentů. V případě, že datový formát neumožňuje vložení časového razítka přímo do dokumentu, dojde k uložení časového razítka do externí struktury. Před expirací platnosti časového razítka proběhne jeho přerazítkování.

Každý typ dokumentu má předdefinovanou dobu uchovávání. Během tohoto období jsou původní data zajištěna řetězcí otisků a časovými razítky. Jednotlivé dokumenty jsou pak automaticky mazány podle definované skartační politiky po uplynutí doby archivace.

3.2.1 Skartace dat

Stejně jako fyzické dokumenty je v některých případech nutné skartovat (např. po uplynutí doby definované zákony, nepotřebná data z podstaty jejich povahy), můžeme skartovat i data elektronická, která jsou uložena v důvěryhodném archivu. Ke skartaci se používá automatická skartační úloha, která maže dokumenty z řetízku, jež jsou starší než počet měsíců X . Automaticky se tedy odmažou všechny dokumenty v řetízku starší než X měsíců podle nastaveného algoritmu:

- spočítá se datum D jako první den v aktuálním měsíci minus X měsíců, (např. je-li dnes 10. 5. 2020 a X je 3, potom $D = 1. 2. 2020$)
- nalezne se nejnovější časové razítko R , pro které platí, že má starší datum než D
- smažou se všechny dokumenty, které mají časové razítko starší nebo rovno R
- smažou se všechna starší časová razítka než R
- systém prokázání původu v čase tímto není narušen, protože inicializační otisk nesmazaného zbytku řetízku je roven otisku, který orazítkovalo razítko R , takže je prokázáno jeho stáří. Tím je zaručeno, že nemůže být libovolně změněn.

Mazání zpráv je transakční na úrovni celé operace, dojde-li tedy při mazání k chybě, celá operace se odrokuje a chyba se zapíše do logu. Veškeré operace jsou zaznamenány v systémovém logu, automatické úlohy navíc vytvářejí protokol o skartaci.

3.3 Věrohodnost dat

Popsané mechanismy, které zajišťují integritu a důvěryhodnost archivovaných dokumentů slouží k prokazování věrohodnosti dat. Součástí archivačních systémů je funkce pro export archivovaných dokumentů na fyzické médium (popř. jako lokální obraz části archivu). Exportovaná data mohou sloužit jednak jako záloha dokumentů, ale zároveň jako podklad pro kontrolní úřady, finanční audity apod. Vyexportovaný obsah spolu se souvisejícími hash řetízky uložený na médiu je podepsán elektronickým podpisem, aby nebylo možné exportovaná data modifikovat. Součástí exportovaného balíčku dat je také offline aplikace pro vyhledávání, náhled a tisk obsažených dokumentů.

Na médiu jsou přiloženy originální soubory s archivovanými dokumenty a část zřetězení otisků těchto dokumentů ohraničená časovými razítky podle zvoleného období. Integritu obsahu média a integritu části zřetězení je možné verifikovat pomocí aplikací, které jsou součástí obsahu média. Pro kontrolu a zobrazení časových razítek se používá aplikace poskytovatele časových razítek, která obsahuje rovněž popis algoritmu použitého pro verifikaci.

Veškeré aktivity probíhající v aplikaci jsou detailně zaznamenány v log souborech a tabulkách aplikace. V případě potřeby mohou být tyto auditní logy archivovány v aplikaci a rovněž zabezpečeny zřetězením jejich otisků a časovými razítky. Tímto způsobem by byla zabezpečena jejich nezměnitelnost.

Součástí důvěryhodného archivu je úloha pro automatickou verifikaci a úloha pro označování zřetězení časovým razítkem. Tyto úlohy probíhají v nastavených intervalech a verifikují integritu všech zřetězení a označují konce zřetězení časovým razítkem nezávislé časové autority.

Archivační řešení obsahuje nadstavbu Business monitoring and reporting popsanou níže, která mimo jiné slouží k zabezpečení a vizualizaci auditní stopy uvedené v aktuálním znění zákona o dani z přidané hodnoty. Jedná se o provozování dokumentů v rámci obchodního případu.

3.3.1 Prokázání důvěryhodnosti dat

Řešení věrohodnosti dat popsané v předešlé kapitole je na území České republiky platné, ověřené auditními a kontrolními úřady a má stejnou váhu jako fyzický doklad. Taktéž splňuje veškeré náležitosti v souladu s legislativou a zákony. Pokud je nutné prokázat důvěryhodnost archivovaných dat, část postupu ověření exportovaných dat by vypadala následovně:

- spuštění verifikace integrity obsahu nad exportovaným vzorkem dat
- spuštění verifikace zřetězení otisků těchto dokumentů
- vyhledání a zobrazením požadovaných dokumentů
- zobrazení časových razítek
- zpřístupnění online aplikace
- vyhledání a zobrazením požadovaného dokumentu
- zobrazení historie průběhu a výsledků automatických úloh razítkování a verifikace zřetězení otisků v daném období
- zobrazení časových razítek
- poskytnutí log souborů s aktivitami, které proběhly v aplikaci v daném období, log soubory jsou archivovány v stejné aplikaci a zabezpečeny zřetězením jejich otisků a opatření časovým razítkem.
- vizualizací auditní stopy, dokumentů provázaných v rámci obchodního případu
- předložení interních bezpečnostních směrnic společnosti pro práci a správu aplikace.

4 ANALÝZA POŽADAVKŮ

Cílem práce je navrhnout a implementovat řešení, které bude dlouhodobě a důvěryhodně archivovat faktury vydané. Tyto faktury vznikají v extérní aplikaci a současně k těmto elektronickým fakturám připojit přílohy související s daným případem. Takto kompletní dokument bude distribuován mezi zákazníky. Žádný jiný dokument se archivovat nebude, řešení není náhradou za DMS systémy.

Cíle projektu by se daly shrnout následovně:

- snížení objemu papírové fakturace, tedy zmenšit náklady na tisk faktur
- splnění požadavků daných legislativou na elektronickou fakturaci
 - prokázání původu účetního dokladu
 - neměnitelnost účetního dokladu
 - čitelnost účetního dokladu
- zrychlit tok dokladů směrem k zákazníkům
- zavést elektronický archiv odběratelských faktur, který nahradí archiv fyzický

Splněním vytyčených cílů projektu budu očekávat následující výstupy:

- kompletace faktur s jejich přílohami
- distribuce faktur s přílohami zákazníkům
- důvěryhodná archivace faktur s jejich přílohami
- webové rozhraní pro přístup a práci s elektronickou fakturací

4.1 Business procesy

Shrnutí business procesu je důležité k pochopení toku dat a následně k nastavení systému podle této logiky. Řešení bude vycházet z obecných postupů pro vystavování dokladů interními systémy (IS, ERP aj.) a k němu se nastaví modul řešící párování faktur s přílohami a následná distribuce.

Nemůžu zde popsat na základě fungování a procesů jednoho systému všechny a tvrdit, že se jedná o obecně platný standard. Mohu ale na základě rozšířených standardů mezi systémové komunikace definovat vstupy do systému a jednotlivé fáze zpracování dat. Proto v navrhnutém řešení budu počítat s následujícími vstupy:

- objednávky, dodávky a faktury budou generované v dohodnutém stylu v interních systémech
- číselník zákazníků bude udržován přímo v archivu
- přílohy budou elektronicky, popř. v papírové podobě (tedy vstup pro další zpracování)
 - elektronické přílohy musí obsahovat rozlišovací znaky v podobě čárového kódu
 - papírové přílohy je potřeba doplnit o rozlišovací znaky v podobě čárového kódu

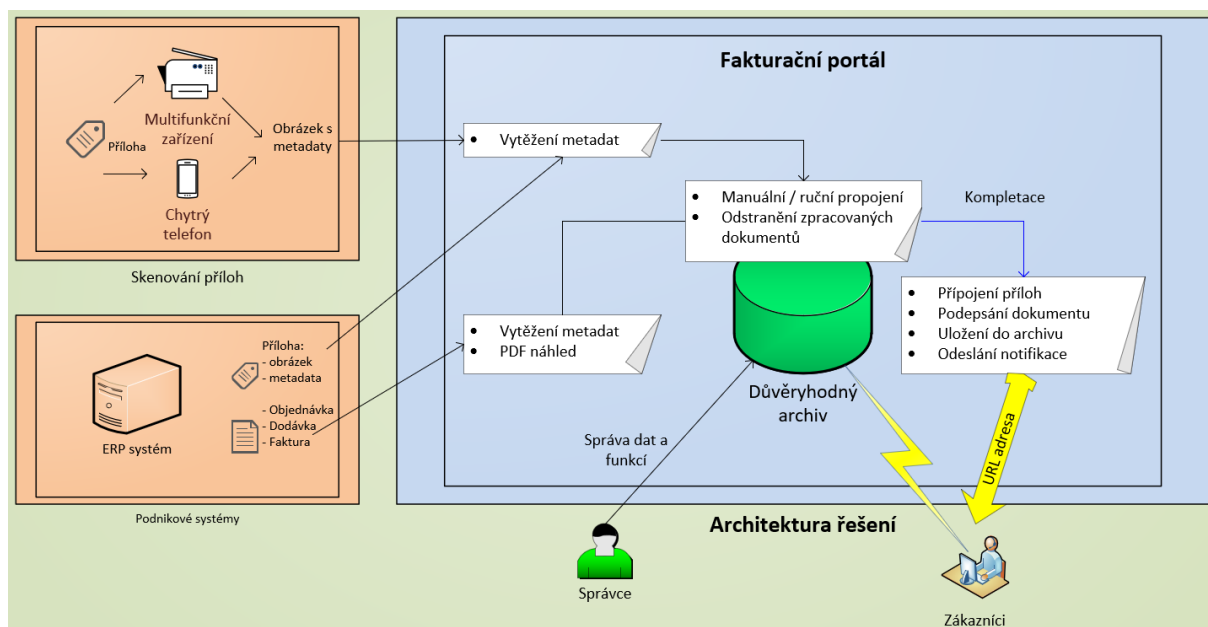
Klíčovým prvkem pro správné fungování a logické propojování faktur s přílohami je nastavení spojovacího (nebo také linkovacího) klíče, kterým faktura pozná její přílohy. Spojovací klíč bude v našem případě číslo objednávky, což je referenční vazba v toku dokladu (Objednávka → Dodávka → Faktura).

Vystavená objednávka, bude mít po odeslání zboží svou dodávku (popř. lze realizovat plnění objednávky více dodávkami), která se po doručení zboží zákazníkovi fakturuje. Faktura tedy obsahuje spojovací klíč v podobě čísla objednávky.

Zpracování faktur bude probíhat v několika fázích:

- příprava faktur (po doručení dodávky) a příloh k načtení do archivu
- kontrola kompletních faktur před distribucí k zákazníkům
- logické propojení faktur se všemi přílohami, dle čísla objednávky
- kompletace faktur, podepsání elektronickým podpisem
- opatření časového razítka na elektronicky podepsanou fakturu
- distribuce faktury k zákazníkům
- uložení faktury do archivu

Obecně fungování archivu popsat způsobem, kdy jsou všechny potřebné dokumenty nahrány na portál (ručně nebo automaticky). Poté obsluha (nebo plánovač) logicky propojí dokumenty dle čísla objednávky. Následně se vygeneruje jediný PDF soubor opatřený digitálním podpisem a odesílá se ven z archivu. Všechny přílohy musí být před začátkem kompletace v archivu. Pokud se nahrají po kompletaci, do výsledného PDF dokumentu se již nepřipojí. Situace popsána obrázkem č. 2.



Obrázek 2: Obecný pohled na fungování systému

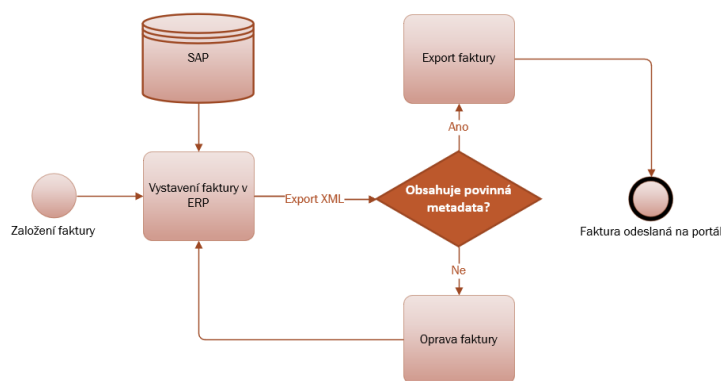
4.2 Příprava faktur

Faktury se připravují v podnikovém informačním systému typu ERP (např. SAP). Takto vytvořené faktury obsahují všechna potřebná metadata a splňují legislativní náležitosti. Procesní postupy, jak se faktury vystavují, nebude práce rozebírat (např. které transakce se používají, která povinná pole jsou nutná pro vystavení faktury apod.), avšak požadavek na export faktur v předem dohodnutém formátu bude muset být ze strany ERP splněn. Strukturu a formáty budou představeny v kapitole č. 5.

Vystavená faktura bude odeslána ze systému ihned po vystavení automaticky do předem specifikovaného místa (v případě firemního prostředí se bude jednat o vzdálené uložště typu FTP, popř. jiné). Specifikem každého systému může být parametrizace na záznamu odběratele, což je pro nás příjemce faktury, aby se vybraným zákazníkům faktura neodesílala. To je však mimo kompetence a oblasti pokrytí archivu.

Proces vytvoření faktury v každém ERP systému počítá s nastavením formuláře na výstupu, tedy grafickou podobou faktury, která je shodná pro všechny zákazníky (může se odlišovat v drobných detailech, jako textové poznámky, hodnoty u jednotlivých položek apod.). Pokud by to bylo možné, je cílem návrhu řešení brát na tento požadavek zřetel a výstupní finální fakturu zachovat ve stejném formátu.

Nedílnou součástí faktury jsou metadata obsahující mimo standardní pole (číslo faktury, příjemce, dodavatel aj.) také čísla objednávek, na jejichž základě je zajištěna vazba na relevantní dokumenty k danému obchodnímu případu (tedy faktuře). Číslo objednávek je povinný údaj, bez kterého nebudou faktury zpracovávány. Informace o distribuci bude portál zjišťovat na základě vlastního číselníku a příjemce faktury. Nebude-li portál obsahovat informace o konečném příjemci faktury, nebude faktura distribuována ven z portálu. Přípravu faktury znázorňuje zjednodušený BPMN diagram (obrázek č. 3).



Obrázek 3: BPMN proces vystavení faktury

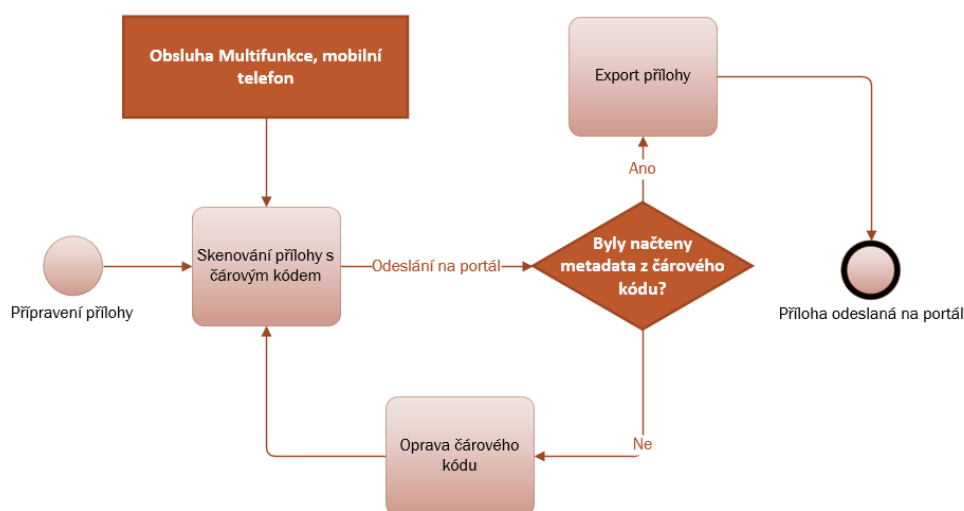
Zde si dovoluji poznámku ke způsobu archivace faktur. Tu můžeme rozlišovat způsobem, zda chceme faktury distribuovat mezi zákazníky, nebo je chceme pouze archivovat bez toho, aby se odesílala notifikace a zákazníci k nim měli přístup. Tohle by měl řešit přímo ERP systém a portál by měl rozhodovat co s fakturou udělat na základě logiky zaslané ERP systémem (např. domluvený příznak ve struktuře XML). Tuhle oblast práce však neobsahuje.

4.3 Příprava příloh

Způsob získávání příloh k fakturám se dá rozlišit dvěma způsoby. První více používaný způsob budou přílohy v papírové podobě. Tyto přílohy budeme muset dostat na portál pomocí skenovacího zařízení. Pro potřeby této práce není žádné technické řešení preferováno a způsob je jen na uživateli. Lze tak přílohy připravit ke zpracování jednak standardním multifunkčním zařízením vybaveném skenerem, nebo můžeme použít fotoaparát mobilních telefonů a přílohu jednoduše vyfotit.

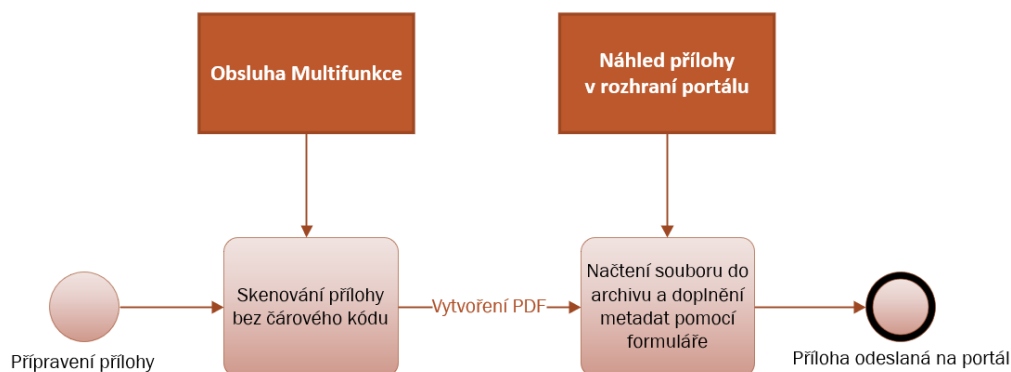
Druhý způsob získávání příloh je přímo ze systému ERP. Víme nyní, že faktura potřebuje spojovací klíč ke svým přílohám, což je číslo objednávky. Objednávku poté odbavíme pomocí dodacího listu (tedy dodávky), která mlže být vyřízena najednou, nebo ve více dodávkách. Takové dodávky jsou také řešeny přímo v ERP, kde vznikají samotné faktury. Proto přidat pomůcku pro kompletaci, tedy čárový kód přímo do formulářů objednávek a dodávek nebude velkým problémem. Způsob, jakým se taková funkcionality implementuje do ERP (např. uživatelský program v SAP generující ke každé dodávce čárový kód, který obsahuje číslo dodávky) v práci řešit nebudu. Pokud je ERP systém schopný vygenerovat do dokladů čárový kód, je schopen přidat informace z tohoto kódu i do metadat, která se zpracují automatickým procesem při vstupu na portál.

Listinné přílohy musí být opatřeny čárovým kódem, který obsahuje metadata potřebná k rozlišení dané přílohy. Čárový kód na přílohy přidává obsluha vystavující faktury, která tak vytiskne a opatří každou přílohu správným štítkem s čárovým kódem. Opět není v rozsahu této práce, aby zmapovala technologie a existující postupy pro tvorbu čárových kódů, ale obecně je dnes velmi mnoho způsobů, jak si vlastní čárové kódy vygenerovat a vkládat do nich potřebné informace (např. online dostupný generátor Barcodes). Samotná tvorba jako i umístění čárového kódu na přílohu je plně v režii zákazníka a ten si musí správné umístění štítků na každou přílohu pohlídat. Proces zachycuje obrázek č. 4.



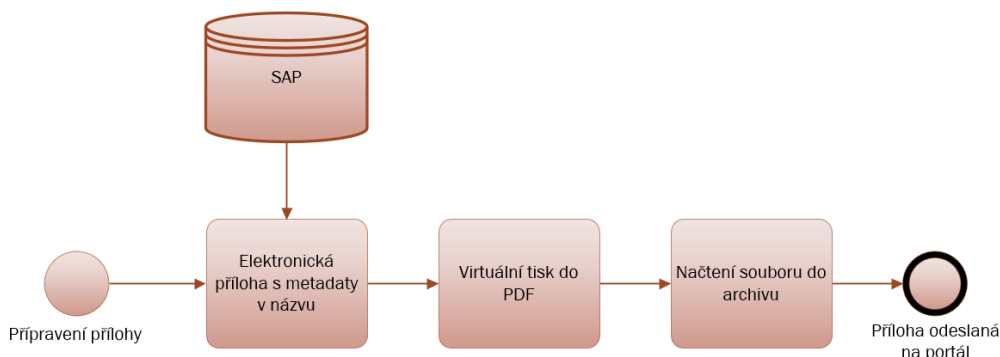
Obrázek 4: BPMN scénář č. 1 pro skenování příloh

Obrázek popisuje scénář, kdy máme přílohu opatřenou čárovým kódem a s jeho pomocí ji nahrajeme na portál. Nicméně existují další scénáře pro přílohy kompletně bez čárového kódu. I na tento případ je brán zřetel a přílohy tak lze k fakturám přidat přímo v prostředí portálu, pomocí ručního nahrání naskenované přílohy (obrázek č. 5). Ta však musí být ve formátu PDF, nikoli v souboru obrázkového typu a metadata se vyplní přímo na portále.



Obrázek 5: BPMN scénář č. 2 pro skenování přílohy

Nicméně existují také scénář pro přílohy opatřené metadaty přímo v ERP systému. Tohle je situace, kdy zdrojová aplikace (ERP) vkládá při exportu (nebo tisku do PDF souboru) čárový kód s informacemi o příloze, které jsou potřebné k jednoznačnému propojení s fakturou. Exportovaný soubor s metadaty se odesílá do sdíleného adresáře, kde dojde ke zpracování portálem (obrázek č. 6).

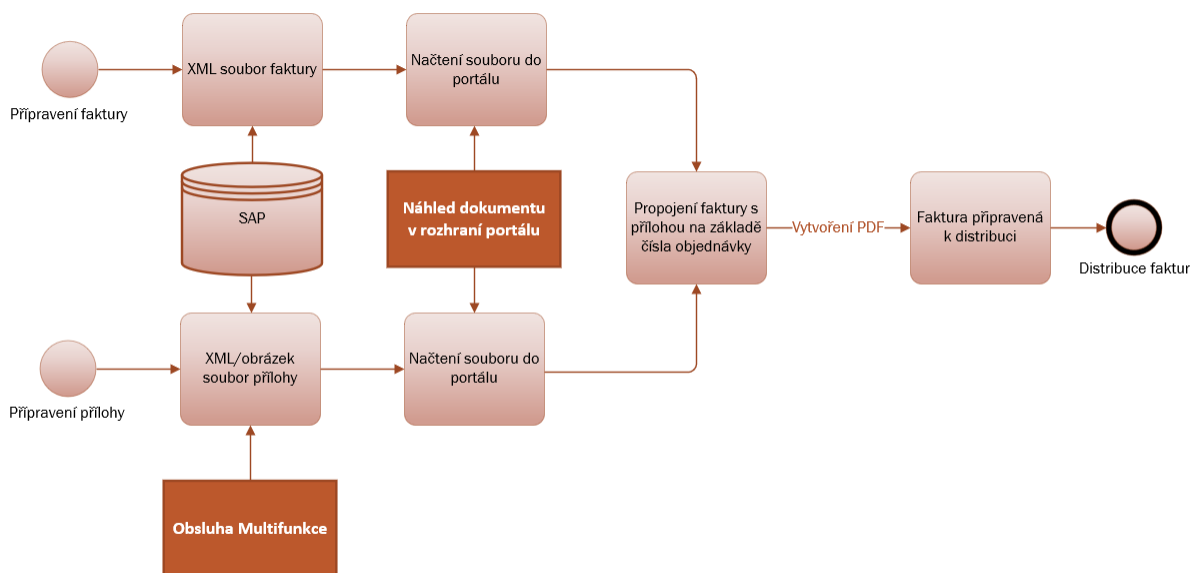


Obrázek 6: BPMN scénář č. 3 pro skenování přílohy

4.4 Kompletace faktur

Jakmile se faktura odešle na portál, je společně se svým obsahem uložena do databáze a pracuje se s ní jako s konceptem. V portálu je dostupná pouze pro administrátora, zákazník nemá sekci s koncepty faktur přístupnou. Na portál jsou následně v pravidelných intervalech zasílány přílohy s připojenými metadaty. Pokud je příloha nahraná ručně uživatelem, je v databázi okamžitě. V opačném případě čeká na zpracování portálem. Jakmile je příloha zpracována (popř. nahrána ručně), lze ji prohlížet, stahovat nebo smazat z portálu. Na základě metadat obsažených v příloze jsou koncepty faktur a přílohy logicky propojeny. Není důležité, v jakém pořadí se dokumenty na portál dostanou (zda bude první faktura, nebo příloha). Avšak v době kompletace musí být na portále již všechny přílohy k dané faktuře. Po doběhnutí zkompletování nebude již možno k faktuře přidat další přílohu.

Celkový zjednodušený proces (neberou se v potaz všechny možné scénáře získávání příloh) od vytvoření faktury po zkompletování je na obrázku č. 7. Kompletace také po doběhnutí zajistí distribuci faktur k zákazníkům.



Obrázek 7: BPMN proces kompletace faktury

5 NÁVRH SYSTÉMU

Cílem této kapitoly bude navrhnout systém portálového řešení pro důvěryhodný archiv faktur, jeho technický koncept, druh komunikace pro výměnu dat mezi systémy apod. Pro jednotné pojmenování a rozpoznání dodaného řešení pojmenujeme požadovaný produkt jako „Elektronická fakturace zákazníkům“ (dále jen archiv EFZ). Finální řešení by mělo tvořit seznam následujících komponent:

- aplikační server, na kterém poběží aplikace
- webové rozhraní fakturačního portálu, které budou používat uživatelé vystavující faktury, ale také zákazníci
- databázový server, který plní funkci fyzického úložiště faktur
- modul portálu pro rozpoznávání s vytěžením čárových kódů z elektronických papírových příloh

Naopak řešení nebude pokrývat následující oblasti, byť některé z nich se dají využít v rámci archivu EFZ částečně, popř. se dá zvolit alternativní řešení:

- multifunkční či jiná skenovací zařízení
- mailový server pro distribuci elektronických faktur k zákazníkům
- službu Microsoft Active Directory pro podporu jednodušší nastavení přístupu uživatelům k archivu EFZ z prostředí organizačních jednotek (např. integrace na SSO)

5.1 Aplikační rozhraní

Je několik způsobu, jak lze nastavit komunikační rozhraní mezi archiv EFZ a podnikové informační systémy (ERP). Lze využít webových služeb nebo strukturované datové formáty (XML, EDI aj.). Výhody a rozdíly mezi těmito způsoby stručně uvedu v této kapitole. Pro archiv EFZ použiji řešení pomocí strukturovaných dat v XML a adresářové uložení. Přenos dat směrem k archivu z integrovaných systémů bude vždy vyvolán akcí na straně ERP. Data se budou zasílat jednotlivě a každý požadavek bude obsahovat jeden dokument a doprovodná metadata.

Konečný datový formát faktur je datový typ souboru PDF, který umožní připojení příloh do datového kontejneru, podepsání elektronickým podpisem a vložení časového razítka. Přílohy se mezi systémy budou přenášet také využitím adresářové struktury.

Implementace rozhraní bude zajištěna v následujícím rozsahu:

- faktury s daty potřebnými ke kompletaci budou ze systému odcházet ve formátu XML
- kmenová data zákazníků budou zakládána ručně v archivu EFZ shodně s daty v ERP
- uživatelské účty budou zakládány ručně v archivu EFZ podle kontaktních osob z ERP evidovaných u každého zákazníka
- ostatní dokumenty relevantní k fakturám budou ze systému odcházet ve formátu XML
- přílohy vytvořené v ERP budou ze systému odcházet ve formátu PDF
- jeden uživatelský účet bude nastaven na jediného zákazníka, tedy v poměru 1:1

5.2 XML komunikace

Komunikace mezi ERP systémem a archivem EFZ bude vystavěna na integrační technologii datových souborů XML. Jedná se o formát dokumentu s libovolně vnořenými elementy, které mohou obsahovat další atributy. Pro pochopení datových formátů XML a jeho struktury, které budu v rámci implementace používat, vysvětlím, co se za pojmem XML skrývá.

Element je uvnitř datové struktury XML specifikován názvem (popř. typem elementu nebo značkou) a v textu je otevřen počáteční značkou (*start tag*). Tento tag je název elementu umístěn mezi symboly < a >. Element ukončujeme koncovou značkou (*end tag*), opět uvozen mezi znaky </ a >. Mezi počáteční a ukončovací značkou je vložen obsah elementu, kterým mohou tvořit dětské elementy a volný text. Obecně platí, že elementy by měly obsahovat data a hodnoty atributu metadata (tedy informace o datech). Struktura souborů používaných v archivu EFZ je definovaná strukturou konkrétních standardů pro dané typy dokladů (objednávka, dodávka a faktura), ale nebudou všechny pole povinné. Podotknu, že implementace plnotučné integrace XML komunikace pro účetní doklady není možné pokrýt rozsahem této práce. Struktury zmíněných dokladů jsou přehledně definované, díky čemuž pro potřeby archivu EFZ mohu vybrat povinná pole splňující nutný základ smysluplné fakturace. [4]

Na základě uživatelské interakce v ERP systému, dojde při zpracování faktury ke spuštění funkce, která nově vytvořenou fakturu pošle na výstup v XML struktuře doplněnou o metadata. Do předem dohodnuté cesty v adresářové struktuře se odešle soubor, kde jej převezme importní služba archivu EFZ. Před samotným importem do archivu EFZ dojde k validaci formátu a obsahu povinných metadat.

Struktura a povinná pole pro faktury ve formátu XML jsou na obrázku č. 8. Nejedná se o kompletní strukturu, ale jen výše povinných polí. Kompletní struktura bude součástí elektronické přílohy pro každý typ XML souboru.

```
<?xml version="1.0" encoding="utf-8"?>
<XmlModel>
  <Invoices>
    <Invoice>
      <InvoiceNumber>FV20XX35897</InvoiceNumber>
      <CustomerId>14</CustomerId>
      <OrderNumber>ORD20XXOU33</OrderNumber>
      <ForeignRate>1</ForeignRate>
      <CurrencyUsed>false</CurrencyUsed>
      <InvoiceDate>2020-05-06T00:00:00</InvoiceDate>
      <InvoiceAddress>
        <Company>Sideshow Collectibles</Company>
        <StreetName>Dukelska</StreetName>
        <StreetNumber>65</StreetNumber>
        <AlternativeStreetNumber>4</AlternativeStreetNumber>
        <City>Karlovy Vary</City>
        <Postcode>36006</Postcode>
      </InvoiceAddress>
    </Invoice>
  </Invoices>
</XmlModel>
```

Obrázek 8: Struktura a povinné segmenty pro faktury

Povinná pole pro fakturaci jsou `<InvoiceNumber>`, `<CustomerId>`, `<OrderNumber>`, `<InvoiceDate>` a vybrané adresní údaje zákazníka `<Company>`, `<StreetName>`, `<StreetNumber>`, `<City>` a `<Postcode>`. Adresní údaje jsou povinné také při zadávání zákazníka přímo do archivu EFZ.

V kapitole 4.1 jsem popsal referenční vazby mezi jednotlivými doklady a logiku pro spojování příloh k jednotlivým fakturám. Pro připomenutí, jedná se o vazbu mezi objednávkou, která má jednu a více dodávek, které se následně fakturují. Tuto referenci si faktura nese v sobě v poli `OrderNumber`. Archiv EFZ potřebuje ke kompletaci i referenční doklady, proto uvedu strukturu a povinná pole pro objednávkové doklady na obrázku č. 9.

```
<?xml version="1.0" encoding="utf-8"?>
<XmlModel>
  <Orders>
    <Order>
      <CustomerId>14</CustomerId>
      <OrderNumber>ORD20XXOU33</OrderNumber>

      <CurrencyUsed>false</CurrencyUsed>
      <OrderDate>2020-05-04T00:00:00</OrderDate>
    </Order>
  </Orders>
</XmlModel>
```

Obrázek 9: Struktura a povinné segmenty pro objednávky

Povinná pole pro fakturaci jsou `<CustomerId>`, `<OrderNumber>`, `<OrderDate>`. Posledním dokladem, se kterým archiv EFZ pracuje je dodávka, jejíž struktura je na obrázku č. 10.

```
<?xml version="1.0" encoding="utf-8"?>
<XmlModel>
  <Deliveries>
    <Delivery>
      <CustomerId>14</CustomerId>
      <OrderNumber>ORD20XXOU33</OrderNumber>
      <DeliveryNumber>9788088152743</DeliveryNumber>
      <DeliveryDate>2020-05-05T00:00:00</DeliveryDate>
    </Delivery>
  </Deliveries>
</XmlModel>
```

Obrázek 10: Struktura a povinné segmenty pro dodávky

Povinná pole pro dodávky jsou `<CustomerId>`, `<OrderNumber>`, `<DeliveryNumber>` a `<DeliveryDate>`. Právě dle hodnoty v poli `DeliveryNumber` se faktury spojují s přílohami do kompletní faktury.

Pokud by v průběhu validace vstupních dat ve formátu XML do archivu nastala chyba (např. chybějící metadata v povinných polích), musí se chyba vyřešit opravou informací přímo ve faktuře ve zdrojovém ERP. Poté zopakovat export faktury.

5.3 Metadata a webové služby

Metadata představují dodatečný soubor informací, který je vytvořen a uložen pro každý dokument. Metadata mohou obsahovat unikátní identifikátor dokumentů, výstavce dokumentů, datum vytvoření a další specifické informace týkající se jednotlivých typů dokumentů a jejich obsahu. V případě strukturovaných formátů dokumentů (např. EDIFACT, XML, ...) je archiv EFZ schopný tato metadata získat z dokumentu zcela automaticky. Pro nestrukturované formáty lze metadata vytvořená uživatelem přiložit k ukládanému dokumentu prostřednictvím webového formuláře v archivu EFZ. Metadata po uložení do databáze slouží zejména jako kritéria pro vyhledávání požadovaných dokumentů.

Pro integraci mezi ERP a archivem EFZ jsem zvolil dostupnější metodu XML datových souborů. Existují však i jiná, efektivnější, avšak robustnější řešení integrace na externí systémy. Jedním z nich mohou být webové služby, kdy spolu komunikují jednotlivé systémy pomocí síťového SOAP protokolu. Pro pochopení této technologie a jak by mohla zefektivnit celý proces komunikace mezi systémy, popíšu modelový případ mezi ERP SAP a obecným webovým portálem.

Komunikace mezi systémem SAP a portálem bude vystavěna na integrační technologii ALE (Application Link Enabling), která je důležitým middleware nástrojem BFA (Business Framework Architecture) architektury. BFA architektura umožňuje SAP i non SAP komponentám od jiných SW výrobců vzájemnou integraci a komunikaci mezi sebou. Data jsou mezi aplikačními systémy vyměňována kontrolovaně a konzistentně, díky asynchronní povaze přenosu data "doputují" do cílového systému, i pokud tento není dostupný v okamžiku jejich odeslání. ALE poskytuje vysoký uživatelský komfort pro administraci komunikace, je možné nastavit opakované zpracování neuskutečněných přenosů, zjistit rychle a přehledně aktuální stav i analyzovat případné chyby.

Pro přenos zpráv se používají speciální datové kontejnery, tzv. *Idocy*. Veškerá přenášená data tak budou kompletně monitorována. V portálu bude SAP systém volat webové služby a výsledek zpracování se pak bude propisovat k jednotlivým Idocům. Vytvoření výstupních Idoců s daty faktury resp. příloh (binární data a metadata) bude integrováno do kopie stávajících tiskových programů. Pro odeslání faktury/příloh bude sloužit nově vytvořená tisková zpráva s tímto nově vytvořeným tiskovým programem a stávajícími formuláři, tím bude možné řídit okamžik odeslání dle specifických potřeb u jednotlivých zákazníků.

Vytváření vstupního Idocu se stavovou informací bude zajišťovat program, který se bude cyklicky dotazovat webové služby na straně portálu na stav doposud nekompletních faktur a pokud došlo k jeho změně, vytvoří Idoc, který zajistí propsání změny do doplňkových dat faktury. Pomocí ALE budou přenášeny rovněž změny odběratelů s využitím standardních SAP nástrojů pro distribuci kmenových dat.

V aplikačním rozhraní webových služeb portálu bude možné volat tyto služby:

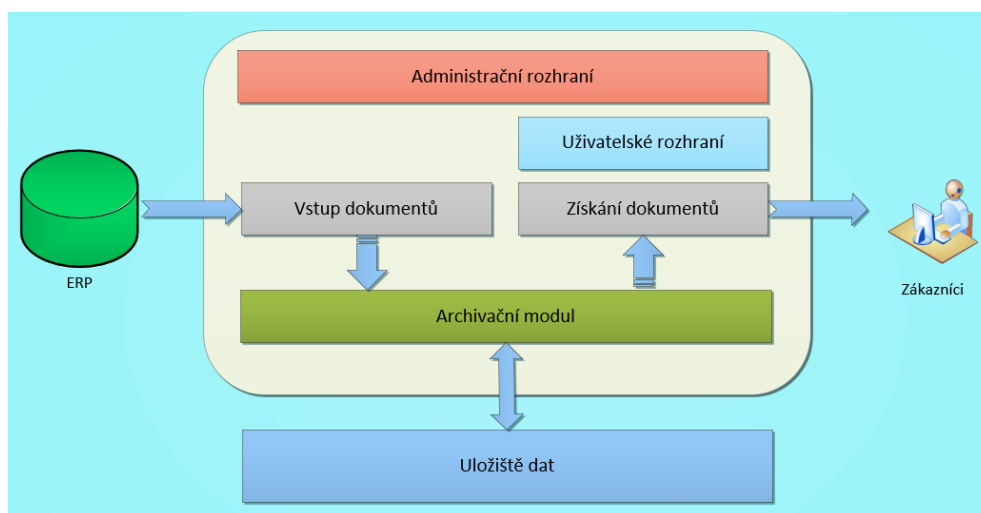
- STORE: Umožňuje uložení konceptu faktury ze SAP nebo alternativní uložení přílohy ze systému zákazníka.
- QUERY: Umožňuje načtení seznamu faktur nebo příloh z portálu na základě zvolených kritérií.
- RETRIEVE: Umožňuje stažení konkrétní faktury nebo přílohy z portálu na základě jedinečného identifikátoru faktury nebo přílohy získaného voláním služby QUERY
- METADATA_UPDATE: Umožňuje aktualizaci metadat faktury nebo přílohy na základě jedinečného identifikátoru faktury nebo přílohy získaného voláním služby QUERY.

Jedná se o modelový případ, jak zefektivní webové služby integrované mezi ERP systémy a portálová řešení komunikaci, zrychlí a automatizují veškeré procesy a rapidně zmenší chybovost, kterou může zapříčinit lidský faktor při ručním zpracovávání dat přímo na portálové aplikaci. Takové řešení nemůže být pokryto v rámci této práce, ale poslouží výborně jako vstupní analýza zadání, existujících řešení a následného vymodelování aktuálního řešení.

5.4 Technický koncept

Archiv EFZ má základní rozhraní pro vstup dat ve formě souborového systému. Z dohodnutého uložistiště bude v pravidelných intervalech načítat data ze strukturovaných XML. Naopak k výstupu z archivu EFZ slouží URL adresa dostupná zákazníkům.

Řešení se skládá ze dvou komponent, aplikačního a databázového serveru. Aplikační server je na logické úrovni rozdělen na administrační a uživatelské rozhraní. Zjednodušený pohled na architekturu celého řešení je na obrázku č. 11.



Obrázek 11: Technický koncept archivu EFZ

6 IMPLEMENTACE SYSTÉMU

Prostudoval jsem teoretickou rovinu problematiky kolem elektronické fakturace a zároveň mám shromážděny veškeré vstupy potřebné pro implementování výsledného produktu. Tato kapitola se bude zabírat praktickou částí práce a seznámí nás s použitými technologiemi, které pro psaní archivu EFZ zvolíme.

Mnohé moduly a části práce jsou závislé na externích, již existujících knihovnách a modulech, které v této kapitole představím, popíšu jejich funkčnost, logiku a následně je zapracuji do archivu EFZ. Pro praktické ukázky budou použity relevantní sekce kódu, které nám danou funkčnost přiblíží. Všechny výseky programových částí pochází z vlastního kódování, nejedná se tak o všeobecné popisy daných knihoven z dostupných zdrojů.

V závěru kapitoly uvedu a stručně popíšu diagram vnitřních procesů, o které se stará integrovaný plánovač. Ten tvoří hlavní funkci archivu EFZ, bez které by neplnil vytyčené cíle. Nemá na starosti pouze jeden proces, ale prozatím tři (všechny tři jsou klíčové a starají se o vlastní část archivu EFZ). Jak uvedu ve shrnutí celé práce, jedná se o nutný základ pro správnou funkčnost. Rozšíření a možnosti dalších směrů jsou nepřeborné.

6.1 Požité technologie

Základem je vývojové prostředí Visual Studio od firmy Microsoft, ve kterém se dobře vyvíjí nejen webové aplikace. Archiv EFZ je napsaný v C# s ASP. NET Core. Aplikace je stejně jako celý původní .NET framework spravována společností Microsoft a byla cílena pouze na operační systém Windows. Aby byla však multiplatformní, přepsal Microsoft původní framework na nový framework .NET Core, který je již multiplatformní, open source a přináší i další výhody (např. vestavěnou Dependency Injection (kód č. 1)), lepší výkon než původní ASP.NET nebo možnost mít více verzí frameworku na jednom stroji najednou. Core používá také modernější architekturu. [5]

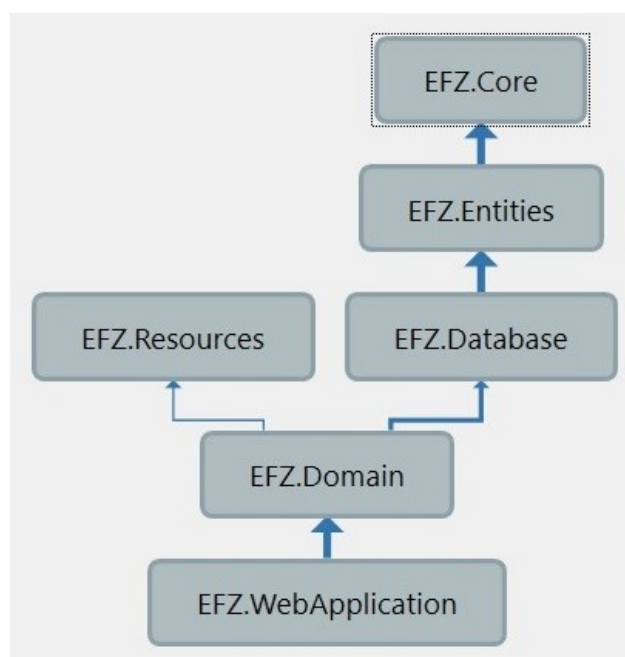
```
services.AddScoped<IInvoiceBlProvider, InvoiceBlProvider>();  
  
services.AddSingleton<IEmailSendGridBlProvider, EmailSendGridBlProvider>();  
  
services.AddSingleton<IAttachmentScanHelperProvider, AttachmentScanHelperProvider>();  
  
services.AddSingleton<IJobSchedulerBlProvider, JobSchedulerBlProvider>();
```

Kód 1: Ukázka vytvoření DI

Technologie Entity Framework (EF) Core ³ je lehká, rozšiřitelná, open source a multiplatformní verze technologie přístupu k datům Entity Framework. EF Core může sloužit jako objekt/relační mapovač (O/RM), který umožňuje vývojářům pracovat s databází pomocí objektů .NET a eliminuje potřebu většiny kódu přístupu k datům, který obvykle potřebují k zápisu. S EF Core se přistupuje k datům pomocí modelu. Model se skládá z tříd entit a objekt kontextu, který představuje relaci s databází, což umožňuje dotazovat a ukládat data. Model se může naimplementovat takovým způsobem, aby odpovídal databázi, nebo ho můžeme z existující databáze vygenerovat.

Archiv EFZ je pro potřeby této práce dostupný na platformě Windows Azure, je tedy dostupný podle URL adresy. Stejně tak lze však k archivu EFZ přistupovat lokálně (příkazem *dotnet run* spustím archiv EFZ v CMD prostředí), popř. se dá hostovat z lokálního počítače, pokud se portál zpřístupní do internetu na dedikovaný aplikační port.

Logika portálu je postavena následujícím způsobem (obrázek č. 12). Webová aplikace odkazuje na doménu (*domain*), ve které je celá logika archivu EFZ. Doména odkazuje na databázi a její entity, což jsou jednotlivé databázové objekty. Každá entita se nahrává do kontextu jako virtuální databázový set. Ke komunikaci s databází slouží jediná třída *CommonDAO()*, která se vkládá do logické business vrstvy.



Obrázek 12: Znáznornění logiky portálu

Správu uživatelů a jejich ověřování má na starosti Microsoftu identity service (user i role jsou identity), která patří pod platformu Azure Active Directory (Azure AD) a umožňuje přihlašovat uživatele k aplikacím. Skládá se z autentizační služby, open source knihoven, registrů aplikace, vývojářské dokumentace a dalších modulů. Specifikuje také jak ukládat uživatele a role (hash hesla, normalizace - metoda *RegistrPassword()*).

³ < <https://docs.microsoft.com/cs-cz/ef/core/> >

Abstraktní rozhraní pro komunikaci s DB zajišťuje DAO (Data access objective) mapováním některých datových operací, čímž nemusím specificky definovat data pro každou entitu (tabulku) zvlášť. Takhle genericky aplikace přistupuje k datům, aniž by potřebovala vstupovat do DB pro konkrétní data. DAO převezmou entitu a spojí ji přímo s databází. Metody komunikující s DB (*GetSingle()*, *GetCollection()*) pro načítání dat z DB (kód č. 2). Další metody komunikující s DB pomocí DAO jsou např. *AddItem()*, *UpdateItem()* pro ukládání dat.

```
public TEntity GetSingle(Func<TEntity, bool> condition, bool withIncludeRelation = true,
    bool asNoTracking = true, ICommonDao<TEntity>.Including including = null)
{
    try
    {
        var databaseContent :IQueryable<TEntity> = GetTable().Cast<TEntity>();
        if (asNoTracking)
            databaseContent = databaseContent.AsNoTracking();

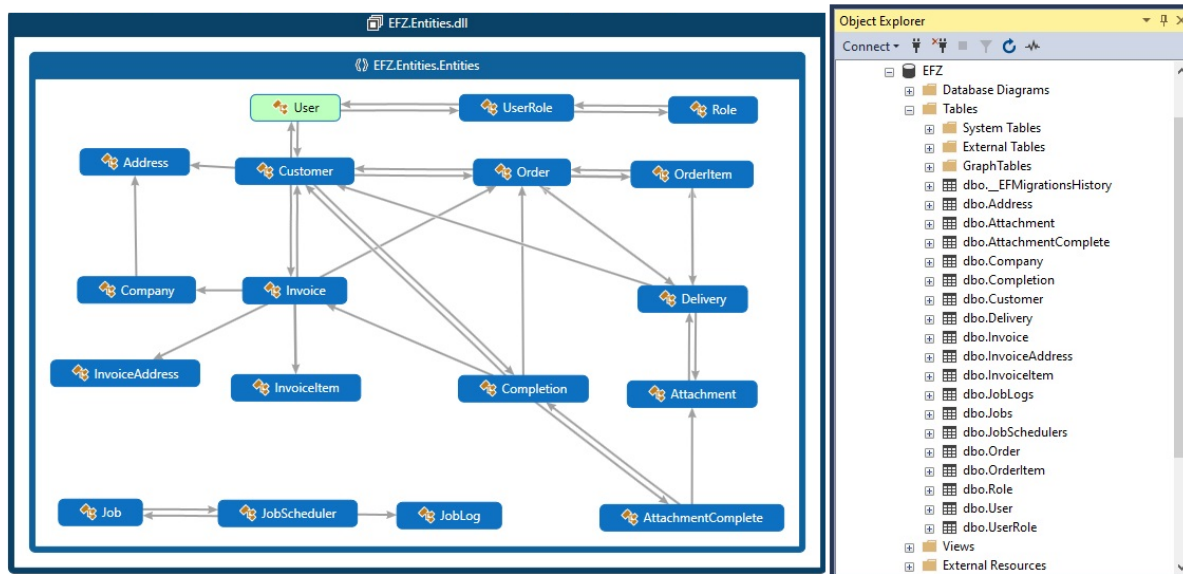
        if (withIncludeRelation)
            databaseContent = GetInclude(databaseContent);

        if (including != null)
        {
            databaseContent = including(databaseContent);
        }
        return !databaseContent.Any() ? null : databaseContent.FirstOrDefault(condition);
    }
    catch (Exception)
    {
        return null;
    }
}
```

Kód 2: Ukázka načítání dat z DB kontextu

6.2 Databázový model

Databázový server je prostředí, kam si archiv EFZ ukládá fyzicky faktury a jejich přílohy. Obsahuje taky veškerou konfiguraci a uživatelské účty. Jako databázový server se používá Microsoft SQL. Návrh a struktura databáze vychází z business požadavků a také vnitřních procesů mezi jednotlivými moduly archivu EFZ.



Obrázek 13: Databázový model archivu EFZ

Celá databáze by se dala rozdělit na dva celky (obrázek č. 13). První zachycuje strukturu a vazby mezi jednotlivými moduly archivu EFZ, tedy cesta, kterou v archivu EFZ dokumenty urazí, než budou zkompletovány. Jedná se o moduly Objednávky (*Order*), Dodávky (*Delivery*), Přílohy (*Attachment*), Faktury (*Invoice*) a Kompletace (*Completion*).

Druhá část zobrazuje meziprocesní komunikaci a vazby mezi jeho částmi. Tohle je systémová část, která je standardním uživatelům nepřístupná. Jedná se o moduly *Job*, *JobScheduler* a *JobLog*.

Návrh databázového modelu si lze prohlédnout včetně parametrizace jednotlivých tabulek v prostředí Microsoft SQL Server Management Studia.

6.3 XML modely

V kapitole popisující návrh systému, jsem definoval pole (respektive jednotlivé tagy) v XML struktuře, se kterými pracuji v archivu EFZ při importu dat. Je však nutné definovat v doméně archivu EFZ samotné modely XML, které bude umět portál zpracovat (kód č. 3).

```

using System.Xml.Serialization;
using EFZ.Entities.Entities;

namespace EFZ.Domain.XmlModel
{
    public class XmlModel
    {
        [XmlArray(elementName: "Invoices")]
        public Invoice[] Invoices { get; set; }

        [XmlArray(elementName: "Orders")]
        public Order[] Orders { get; set; }
        [XmlArray(elementName: "Deliveries")]
        public Delivery[] Deliveries { get; set; }
    }
}

```

Kód 3: Ukázka definice používaných XML modelů

Při importu se kontroluje klíčový prvek, což je číslo objednávky (*OrderNumber*). Pokud by při importu číslo již existovalo, skončí import dat chybou a faktura se do archivu EFZ nepřenesení. Informace o neúspěšném importu je dostupná v logu jednotlivých procesů. Platí i pro import dat pro sekci Objednávky a Dodávky v archivu EFZ (kód č. 4).

```

if (xml.Invoices != null)
{
    foreach (var invoice in xml.Invoices)
    {
        if (invoiceDao.GetSingle( condition: t => t.OrderNumber.Equals(invoice.OrderNumber)) != null)
        {
            jobLogs.Add( item: GetLogException(fileName, new Exception( message: $"Invoice with order number:{invoice.OrderNumber} existing.")));
            continue;
        }
        var orderId :long? = orderDao.GetSingle( condition: t => t.OrderNumber.Equals(invoice.OrderNumber), withIncludeRelation: false)?.Id;
        invoice.OrderId = orderId;
        invoice.InvoiceItems = invoice.XmlInvoiceItems?.ToList();
        invoiceDao.AddItem(invoice);
    }
}

```

Kód 4: Ukázka importu XML dat pro vytvoření faktury

6.4 DinkToPdf

DinkToPdf - open source ⁴ knihovna, která umožňuje konvertovat HTML do PDF dokumentu. Dokument musí být správně definován, aby se konverze povedla (kód č. 5). Definice ve smyslu jednotlivých segmentů ve výsledném dokumentu (velikost formuláře na A4, orientace papíru, barevné schéma, kódování znaků, velikost a typ fontů a další parametry). [6]

⁴ Open source – jedná se o počítačový software s otevřeným kódem. Ten je dostupný technicky, ale i licenčně.

```

public byte[] GetInvoicePdfDocument(Invoice invoice)
{
    var globalSettings = new GlobalSettings
    {
        ColorMode = ColorMode.Color,
        Orientation = Orientation.Portrait,
        PaperSize = PaperKind.A4,
        Margins = new MarginSettings {Top = 10},
        DocumentTitle = $"Faktura - {invoice.InvoiceNumber}"
    };
    var objectSettings = new ObjectSettings
    {
        PagesCount = true,
        HtmlContent = HtmlStringBuilder.HtmlInvoiceDocument(invoice),
        WebSettings =
        {
            DefaultEncoding = "utf-8",
            UserStyleSheet = Path.Combine(Directory.GetCurrentDirectory(), "wwwroot", "css/site.css")
        },
        HeaderSettings = {FontName = "Calibri", FontSize = 10, Right = "Page [page] of [toPage]", Line = true},
        FooterSettings =
        {
            FontName = "Calibri", FontSize = 10, Line = true, Center = $"Faktura - {invoice.InvoiceNumber}"
        }
    };
    var pdf = new HtmlToPdfDocument()
    {
        GlobalSettings = globalSettings,
        Objects = {objectSettings}
    };
}

```

Kód 5: Ukázka vytvoření PDF faktury

Pro úspěšné vytvoření PDF dokumentu, je nutné nejdříve vytvořit fakturu jako HTML řetězec, který se následně konvertuje do PDF. Tohle provádí metoda *GetInvoicePdfDocument()*, která importovaná data z XML zdroje převede z HTML řetězce do PDF souboru, který se bude následně kompletovat.

6.5 iTextSharp

Knihovna, která umožňuje vytvářet, upravovat a kontrolovat PDF dokumenty ⁵. Jedná se o port původní knihovny iText pro Javu. Nejedná se o *fork*, vývoj stále probíhá v Jave a k jednotlivým verzím se vydávají jejich .NET porty. Knihovna umí generovat i jiné soubory, než PDF (např. RTF nebo HTML). Strukturu vytvořeného PDF souboru lze exportovat do XML (samotný soubor obsahuje metadata), s čímž se dá nadále pracovat. Pokud by se zapojil HTML parser, lze vygenerovat PDF soubor přímo z HTML. [7]

Metoda *CompletingPdf()* (kód č. 6) zpracovává kompletaci finálního PDF dokumentu, tedy na základě spojovacího klíče (OrderNumber) doplní k faktuře veškeré přílohy a následně celý dokument digitálně podepíše.

⁵ < <https://itextpdf.com/en> >

```

var invoiceIncluded = invoiceDao.GetSingle( condition: t => t.Id.Equals(invoice.Id));
var invoicePdfDocumentArray :byte[] = GetInvoicePdfDocument(invoiceIncluded);
var attachmentFiles :List<FileStream> = GetAttachmentsFile(attachments);

if (!Directory.Exists(path)) Directory.CreateDirectory(path);
CompletionPdf(invoicePdfDocumentArray, attachmentFiles, fileName: $"{path}{serverFilename}");

if(attachmentFiles!=null && attachmentFiles.Any())
    attachmentCompletes.AddRange( collection: attachments.Select(t => new AttachmentComplete(){AttachmentId = t.Id}));

var completion = new Completion()
{
    CustomerId = invoice.CustomerId,
    InvoiceId = invoice.Id,
    OrderId = invoice.OrderId,
    OrderNumber = invoice.OrderNumber,
    InvoiceNumber = invoice.InvoiceNumber,
    InvoiceDate = invoice.InvoiceDate,
    CompleteFileName = $"{invoice.InvoiceNumber}.pdf",

```

Kód 6: Ukázka kompletace PDF dokumentu

Proces kompletace vyčítá data z modulu Faktury podle příznaku „*IsCompleted*“ (kód č. 7). To je v entitě datový typ *boolean* (true, false). Pokud faktury jsou zkompletovány, tak jdou na další. Pokud nejsou (dle hodnoty), podívám se do příloh, zda čeká nová příloha na kompletaci a potom kompletuji. Parametrem *CompletionCounter()* lze nastavovat počet pokusů kompletace, což se využije ve chvíli, kdy nemám všechny relevantní přílohy na portále (aktuálně nastaveno pro dva pokusy).

```

};
completionDao.AddItem(completion);
invoice.IsCompleted = true;

foreach (var attachment in attachments) attachment.IsCompleted = true;

var deliveries :IEnumerable<Delivery> = deliveryDao.GetCollection( condition: t => t.OrderNumber.Equals(invoice.OrderNumber));
foreach (var delivery in deliveries)
{
    delivery.IsCompleted = true;
}

```

Kód 7: Příznaky na faktuře pro kompletaci

6.6 Digitální podpis

Funkce archivu EFZ, která umožňuje podepisovat PDF dokumenty, čímž se prokáže původ dokumentů a zároveň zkompletované dokumenty opatří časovým razítkem nezávislé certifikační autority. Nejdříve se načtou data certifikátu pro podpis, který musí obsahovat privátní klíč (certifikát je uložen v souboru s příponou pfx ve formátu PKCS #12). Při načtení musí mít program dostupné i heslo certifikátu (kód č. 8). Certifikát se načte aplikací ve formě *FileStreamu*. Lze však pracovat i s certifikáty v lokálním uložišti systému Windows.

Z certifikátu se získá privátní klíč, který je uložený v proměnné *key* typu *ICipherParameters* a cesta k certifikátu - *certificate chain* typu *X509Certificate[]*. Obě tyto proměnné jsou použity v metodě *SetCrypto()* - objektu *SignatureAppearance*, která klíč přiřadí PDF dokumentu.


```

var sap:PdfSignatureAppearance = st.SignatureAppearance;

var cert = new Cert(path: "EFZ.pfx", password: "psw");

sap.SetCrypto(cert.Akp, cert.Chain, crlList: null, PdfSignatureAppearance.SelfSigned);
sap.Reason = "Archived digital signature";
sap.Contact = "EFZ";
sap.Location = "EFZ";

sap.CertificationLevel = PdfSignatureAppearance.CERTIFIED_NO_CHANGES_ALLOWED;

```

Kód 8: Ukázka načtení (EFZ.pfx) podpisového certifikátu

PDF stamper je plugin, který umožňuje dynamicky podepisovat PDF dokumenty údaji, které jsou obsaženy v certifikátu (např. jméno, e-mail, adresa). Komponenta může rovněž k PDF dokument orazítkovat obrázky (JPEG, PNG, TIFF a GIF), zvládne také vypsat informace z metadat nebo tato data aktualizovat. Komponentu lze jednoduše integrovat do aplikací, které podporují ActiveX komponentu.

Metoda *GetTimeStampToken()* (kód č. 9) přidá do finálního PDF dokumentu časový token (tedy časové razítko). Jelikož ho vystavuje nezávislá autorita (u nás je to akreditovaná První certifikační autorita, a.s. (I.CA)), tak splňuje legislativní požadavky a garantuje tak důvěryhodnost dokumentu v čase. Metoda, zda mít certifikáty dva (jeden pro podpis, druhý pro časové razítko) nebo jeden je jen o nastavení systému. Pro jednoduší správu a hlídání termínu vypršení platnosti certifikátu (časová razítka mají platnost rok) je efektivnější používat certifikát jeden.

```

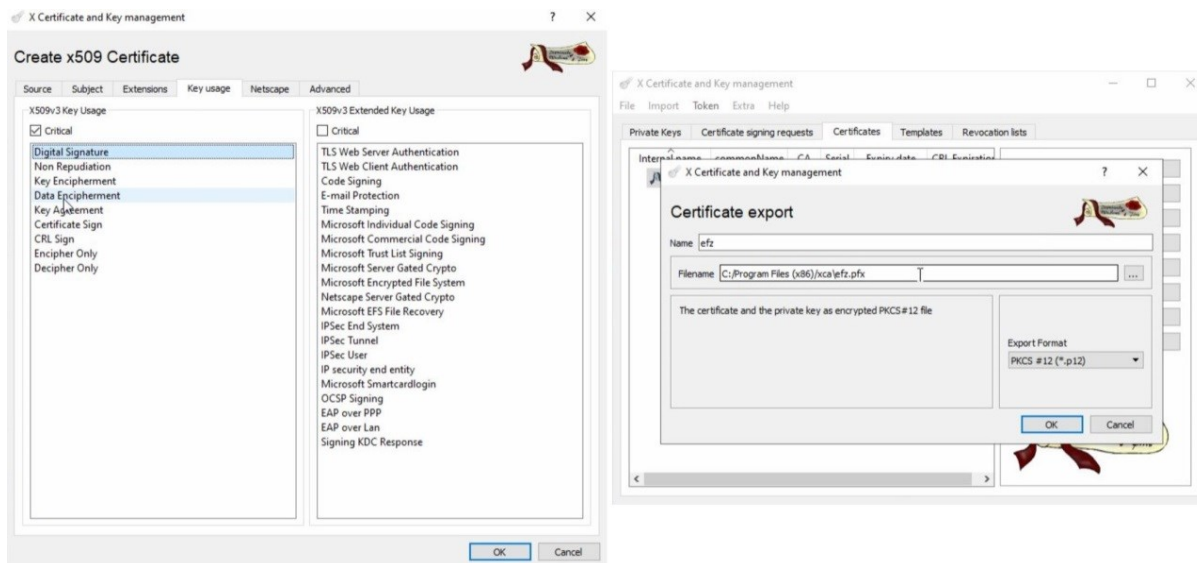
byte[] tsImprint = new byte[messageDigest.GetDigestSize()];
messageDigest.DoFinal(tsImprint, outOff: 0);
var cal:DateTime = DateTime.UtcNow;
byte[] tsToken = tsa.GetTimeStampToken(caller: null, tsImprint);

byte[] ocsp = null;
if (cert.Chain.Length >= 2)
{
    String url = PdfPkcs7.GetOcspurl(cert.Chain[0]);
    if (url != null && url.Length > 0)
        ocsp = new OcspClientBouncyCastle(checkCert: cert.Chain[0], rootCert: cert.Chain[1], url).GetEncoded();
}
byte[] sh = sgn.GetAuthenticatedAttributeBytes(secondDigest: tsImprint, cal, ocsp);
sgn.Update(sh, off: 0, len: sh.Length);

```

Kód 9: přidání časového razítka

Pro účely testování postačuje vygenerovat vlastní certifikát, např. integrovanými funkcemi v OS Windows Serveru, nebo použít open source aplikace (obrázek č. 14). Certifikát použitý pro archiv EFZ byl vygenerován pomocí nástroje XCA (X - Certificate and Key management).



Obrázek 14: Aplikace XCA pro generování certifikátu

V případě potřeby lze v dokumentu vizuálně zobrazovat certifikát, k čemuž se použije metoda *SetVisibleSignature()* společně s parametrem, kde se certifikát v dokumentu umístí.

6.7 Free Spire.Barcode

API pro práci s čárovými kódy vyvinuté pro .NET, které dokáže generovat a číst 1D i 2D (QR) čárové kódy. Relativně snadně lze integrovat zpracovávání čárových kódů do aplikace, která tak bude schopna zpracovat mnohé formáty obrázků (BMP, JPG, PNG, EMF, TIFF, GIF aj.). Existuje i komerční verze, která je však zpoplatněna. Ke splnění jednoho z hlavních cílů této práce však Free verze fungovala. [8]

Princip komponenty spočívá v naskenování obrazového dokumentu (tedy obrázek libovolného podporovaného formátu) a jeho konverze do PDF. Během konverze je obsah převeden do datového řetězce, který se prohledává za účelem nalezení datového řetězce (string). Pokud nějaký řetězec najde, prohledá v databázi záznamy „*deliveryNumber*“.


```

public List<string> GetBarcodeFromStream(Stream stream)
{
    var results = new List<string>();
    var reader :string[] = Spire.Barcode.BarcodeScanner.Scan(stream, IncludeChecksum: true);

    if (reader != null)
    {
        results.AddRange(reader);
    }

    return results;
}

```

Kód 10: Ukázka čtení čárových kódů z obrázku

Samotná komponenta „*BarcodeScanner()*“ (kód č. 10) prochází načtený stream a skenuje to na všechny typy čárových kódů, ze kterých vrátí seznam všech stringů, které našel a byl schopný vyčíst (kód může být poškozen, špatně čitelný, apod.). Prochází celou stranu, nemá definované konkrétní místo. Bude-li vstup obsahovat více stran, bude procházet každou zvlášť. To se však promítne znatelně do výkonu (řádově minuty na každou stranu).

Metoda *GetBarcodeFromStream()* (kód č. 11) nemusí vždy najít kód ze streamu (hodnota null). V takovém případě skončí konverze souborů chybou (výjimky na konci kapitoly) a příloha se neuloží. Pokud najde shodu, tak převede vstup do PDF souboru, vytvoří záznam v DB (*new Attachment()*), ke kterému přiřadí číslo objednávky (spojující klíč *OrderNumber*), soubor pojmenuje a skenuje dále.

```

{
    var deliveryNumbers :List<string> = _attachmentScanHelper.GetBarcodeFromStream(stream);
    var deliveryDao = new BaseDaoFactory( context:new EfzDbContextFactory().Create()).GetDao<Delivery>();
    foreach (var deliveryNumber :string in deliveryNumbers)
    {
        var delivery = deliveryDao.GetSingle( condition: t :Delivery => t.DeliveryNumber.Equals(deliveryNumber));
        if (delivery != null)
        {
            deliveryNumberString = delivery.DeliveryNumber;
            var pdfStream = _attachmentScanHelper.ConvertImageToPdf(stream);
            var entity = new Attachment()
            {
                OrderNumber = delivery.OrderNumber,
                DeliveryId = delivery.Id,
                DeliveryNumber = delivery.DeliveryNumber,
                FileName = $"{fileName}.pdf"
            };

            UploadAttachment(pdfStream, entity);
            status = true;
            break;
        }
    }
}

```

Kód 11: Ukázka prohledávání DB dle vyčtených záznamů

Vyčítání datového streamu, může skončit díky jedné ze dvou chyb (*Exception*). Program buď nenajde žádný čárový kód ve zdroji, nebo se nalezený řetězec neshoduje s žádným záznamem v DB (kód č. 12).

```
stream.Dispose();
if (status)
{
    if (File.Exists(filePath)) File.Delete(filePath);
    jobLogs.Add(item: GetLogException(fileNameWithExtension, new Exception(message: $"Add as attachment for delivery: {deliveryNumberString}")));
}
else
{
    jobLogs.Add(item: GetLogException(fileNameWithExtension, new Exception(message: "Barcode not exist or not match to any delivery number.")));
}
```

Kód 12: Ukázka úspěchu či neúspěchu vytěžení dat

6.8 PDF Sharp

Další open source použitá knihovna, která snadno vytváří PDF dokumenty z obrázku na vstupu. Pokud je nalezena shoda se záznamy v DB, uloží se nalezená hodnota k dokumentu ve formě metadat. Následně metoda „*ConvertImageToPdf()*“ převede skenovaný obrázek do PDF souboru (kód č. 13). [9]

```
public MemoryStream ConvertImageToPdf(FileStream stream)
{
    var resultStream = new MemoryStream();
    var doc = new PdfDocument();
    doc.Pages.Add(new PdfPage());
    var xgr :XGraphics = XGraphics.FromPdfPage(doc.Pages[0]);
    var img :XImage = XImage.FromStream(stream);

    xgr.DrawImage(img, xi:0, yi:0);
    doc.Save(resultStream);
    doc.Close();
    return resultStream;
}
```

Kód 13: Ukázka konverze obrázku do PDF

6.9 SendGrid

Cloudová služba Microsoftu, která poskytuje spolehlivé transakční doručování e-mailů ⁶ (jedná se o typ e-mailu odesílaného k usnadnění dohodnuté transakce mezi odesílatelem a příjemcem), škálovatelnost a analýzy v reálném čase spolu s flexibilními API ⁷, která usnadňují integraci do vlastních projektů. Služba se běžně využívá pro správu pravidelných rozesílání propagačních akcí dle distribučních seznamů, shromažďování metrik v reálném čase, e-maily pro vytváření nových účtů, resetování hesla apod (kód č. 14).

⁶ < <https://sendgrid.com/> >

⁷ API (Application Programming Interface) v softwarovém inženýrství označení používané pro rozhraní používané k programování aplikací.

```

public EmailSendGridBIPProvider(IConfiguration configuration)
{
    _configuration = configuration;
}

public void SendInvoiceCompleteNotification(Completion completion)
{
    var apiKeyString = _configuration.GetSection("key: SENDGRID_API_KEY").Value;
    var client = new SendGridClient(apiKey);
    var from = new EmailAddress("no-reply@efz.cz", "EFZ - Elektronická fakturace zákazníkem");
    var urlString = string.Format(LabelsMessages.valcompletionurl, completion.Id);
    string subject, htmlContent;
    var recipients = new List<EmailAddress>();
    if (completion.Customer?.Users != null)
        recipients.AddRange(collection completion.Customer.Users.Select(t => new EmailAddress(t.Email, t.Name)).ToList());

    subject = $"Nová faktura: {completion.OrderNumber}";
    htmlContent = string.Format(LabelsMessages.valcompletionContent,
        completion.OrderNumber,
        url, LabelsMessages.valcompletionurlName);

    var plainText = "";
    var msg: SendGridMessage = MailHelper.CreateSingleEmailToMultipleRecipients(from, recipients, subject, plainText, htmlContent);
    client.SendEmailAsync(msg);
}

```

Kód 14: Ukázka rozesílání e-mailových notifikací

Metoda *SendInvoiceCompleteNotification()* rozesílá notifikaci zákazníkům po kompletaci faktury. Jakmile je kompletace hotova, odešle se na registrovaný e-mailový účet zákazníka zpráva o nové fakturě č. XY (archiv EFZ doplní XY dle čísla objednávky). Odesílatel je informační schránka archivu EFZ (no-reply@efz.cz). Tělo zprávy obsahuje pouze odkaz na stránky archivu EFZ, nikoli přiloženou fakturu. Zákazník se tedy jediným klikem dostane přímo z mailu na svůj účet, kde si fakturu může prohlédnout, popř. stáhnout. Podoba informačního mailu bude zobrazena v kapitole o funkcích archivu EFZ.

K usnadnění kódu slouží operátor pro šíření nulové hodnoty (*Customer?.Users != null*). Ten usnadňuje zpracování hodnoty *null* v řetězci volání metod a zastavuje volání v případě nálezu první *null* hodnoty.

6.10 Diagram - JobSchedulerBIPProvider

Diagram znázorňuje strukturu vazeb mezi časovačem všech úloh a jeho jednotlivými procesy. Každý ze tří procesů má definované jednotlivé metody, které vykonává. Větší detail lze vygenerovat přímo z Visual Studia a lze si tak zobrazit detailní pohled na obsahy jednotlivých tříd a jejich parametry. Diagram slouží pro vizualizaci komplexnosti vazeb mezi vnitřními procesy archivu EFZ.

SetupXmlJobTimer – Proces importu dat z předem připravených XML datových souborů

SetupAttachmentScanJobtimer – Proces zpracování příloh a vytěžení dat z čárového kódu

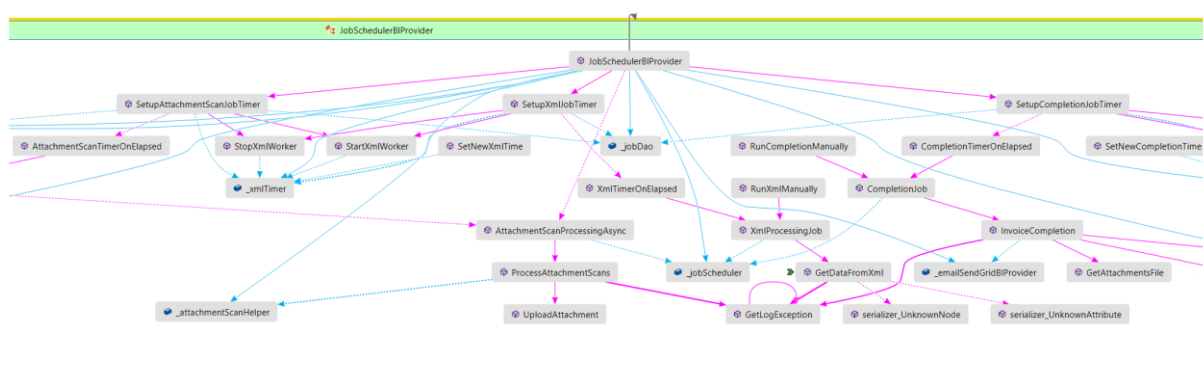
SertupComplettionJobTime – proces kompletace faktur s přílohami

Pro všechny procesy definují nastavením, kdy se daný proces spustí, popř. zda se vůbec proces spustí. V tom případě se jedná o manuální spouštění. Časovač (*timer*), který definuje časy je pro každý proces jiný:

- *_xmlTimer*
- *_attachmentScanTimer*
- *_completionTimer*

Časovač však může definovat dobu běhu, po které se proces spustí. V takovém případě je proces spuštěn po vypršení nastaveného času:

- *XmlTimeOnElapsed*



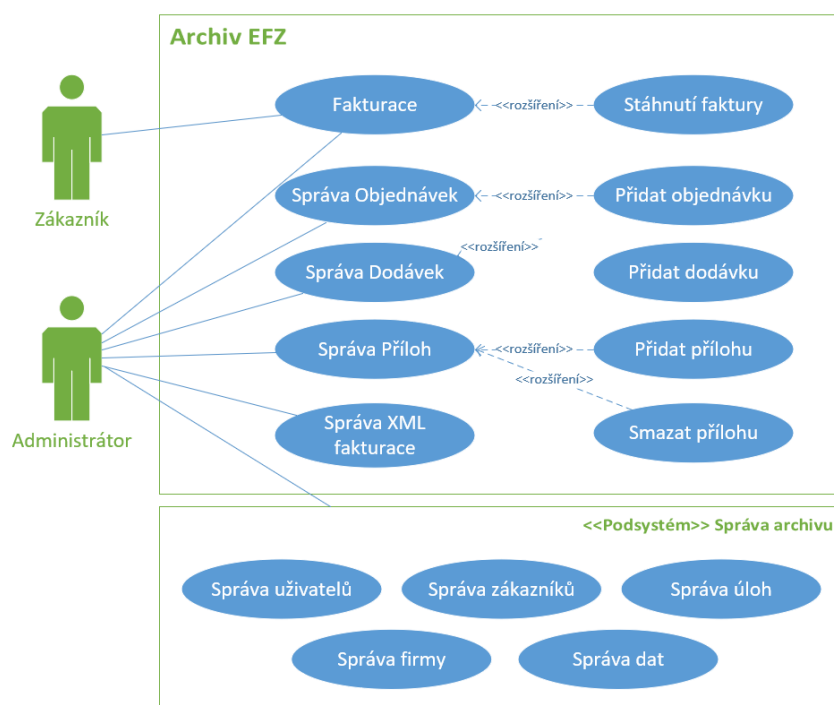
Obrázek 15: Nekompletní diagram Časovače úloh

Část diagramu s procesy a jejich metodami je na obrázku č. 15. Z důvodu objemnosti diagramu, nešlo zachytit jinou výšeč, která by šla na malém prostoru lépe vizualizovat. Kompletní diagram je přiložen v elektronické podobě spolu se všemi kódy této práce.

7 FUNKCE ARCHIVU

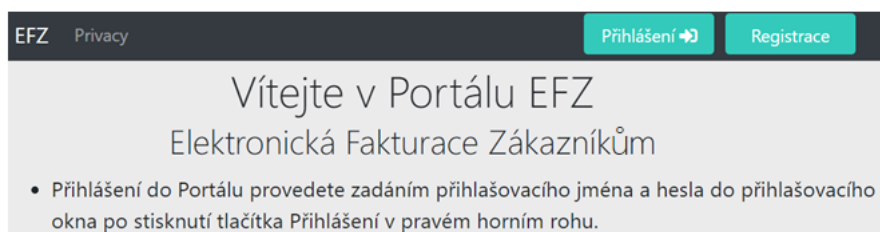
Archiv EFZ je nyní naimplementován, obsahuje testovací data a lze ho nyní představit z pohledu koncového uživatele, jehož role a oprávnění je v archivu omezeno na minimum. Proti tomu role správce nemá žádné limity a uživatel s touto rolí má kompletní správu nad funkcemi archivu. Pro vlastní práci je archiv dočasně dostupný na URL adrese <https://efz.azurewebsites.net/>

Základní pohled na interakci uživatelů se systémem je zachycen na obrázku č. 16. Nejedná se o kompletní funkcionalitu, ty budou popsány podrobněji v následujících kapitolách (role uživatele a administrátora).



Obrázek 16: Základní diagram užití systému

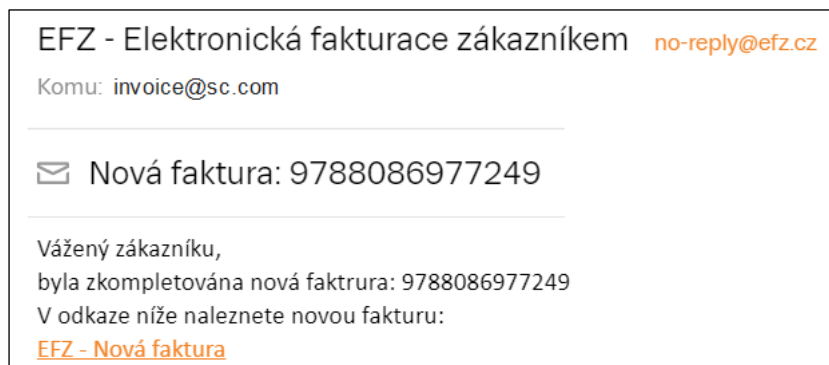
Domovská stránka portálu nabídne pouze informace, jak se přihlásit do zákaznického účtu (obrázek č. 17), pokud účet není založen, je uveden postup pro založení nového účtu. V případě problému je uveden kontakt na administrátora archivu EFZ. Součástí veřejné části archivu EFZ je informativní sekce o zpracovávání osobních údajů v rámci portálu EFZ (Privacy Policy).



Obrázek 17: Domovská stránka archivu EFZ

7.1 Uživatelské rozhraní

Archiv EFZ zajistí rozesílání kompletních faktur zákazníkům. E-mail neobsahuje fakturu ve formě přílohy, ale obsahuje odkaz (obrázek č. 18) pro přístup na portál do zákaznické sekce společně s číslem nové faktury, která byla archivována. Příjemce zprávy se tak může přímo přes odkaz ve zprávě připojit na portál použitím jména a hesla. Zde má již přístup ke všem svým fakturám, které si může kdykoli stáhnout z archivu EFZ a vytisknout. Žádné další operace nemůže zákazník uskutečnit.

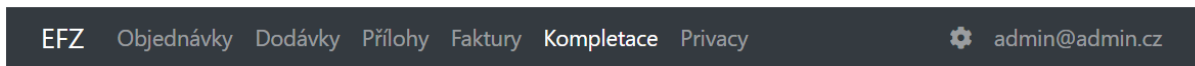


Obrázek 18: Text informačního e-mailu o nově vystavené faktuře

7.2 Administrace EFZ

Uživatel s oprávněním administrátora má úplnou kontrolu nad archivem a jeho funkcemi. Z praxe je vhodné administrátorský účet obecně nepoužívat a jeho oprávnění přiřadit konkrétnímu uživateli. Není to jen z důvodu transparentnosti (obecně pod administrátorským účtem nezjistíme, kdo mohl dané operace vykonat), ale i v případě archivu EFZ, který musí být v souladu s legislativou, je to nutné.

Po přihlášení do systému má administrátor přístupné všechny moduly systému, kdežto uživatel má přístup pouze do Fakturace. U administrátora není sekce Faktury obsahově shodná s tím, co vidí uživatel. Administrátor má v sekci Faktury nekompletní dokumenty, které se budou procesem kompletovat s přílohami a následně podepisovat certifikáty. Dokumenty, které má přístupné uživatel, má administrátor v sekci Kompletace (obrázek č. 19).



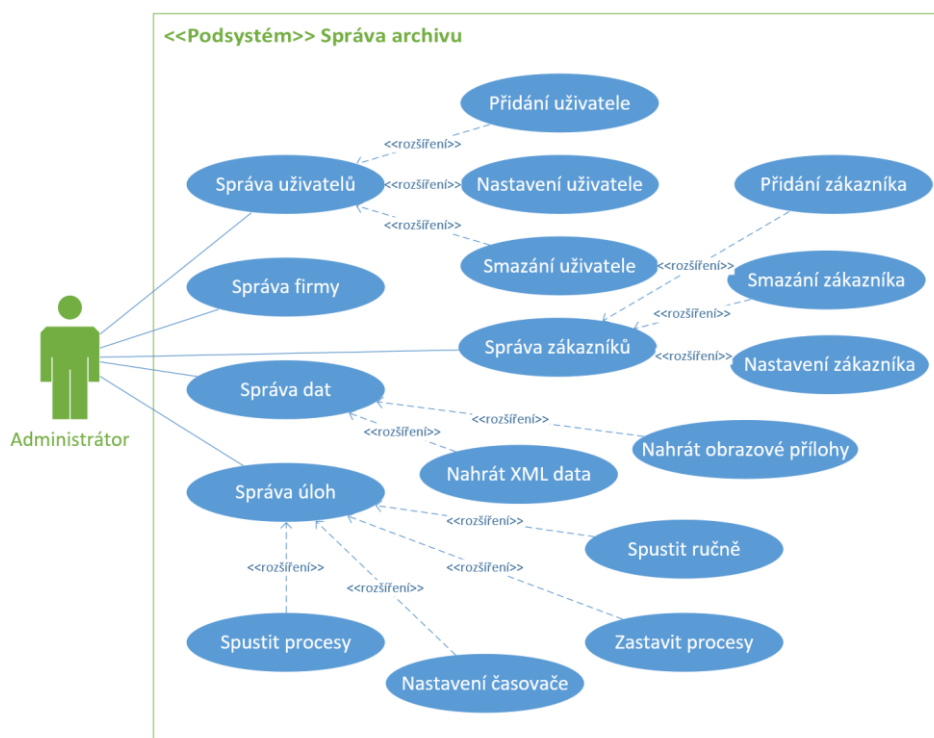
Obrázek 19: Kompletní sekce archivu EFZ

V modulu Fakturace ani administrátor nemá povoleny žádné aktivity. Dostupný je pouze náhled na nainportované faktury z XML dat obdržených na vstupu archivu EFZ. Prohlízet data z XML vstupu jde také v sekci Objednávky a Dodávky s tím rozdílem, že data lze zadat v těchto sekcích i ručně. Dříve jsme si řekli, že faktury se linkují dle spojovacího čísla objednávky, která má následně jednu a více dodávek. Objednávka se poté uvádí ve fakturaci (jako referenční doklad), což

je potřebný klíč pro kompletaci. Pokud bychom tedy chtěli fakturovat a neměli v systému žádnou objednávku s dodávkou, může je administrátor zadat pomocí formuláře ručně. Formuláře pro novou objednávku a dodávku jsou velmi podobné, povinné parametry je nutné vybrat z otevíracích seznamů. Přidání potřebných záznamů je tak velice intuitivní. Ruční vložení přílohy se liší v jednom kroku, kdy se musí vybrat PDF soubor uložený např. v lokálním uložišti (obrázek č. 20).

Obrázek 20: Formulář pro ruční nahrání přílohy

Pouze administrátor má u svého jména ikonu pro vstup do podsystému, kde probíhá veškerá správa archivu EFZ a jeho funkcí. Přehled funkcí pro správu archivu je zobrazen na obrázku č. 21.



Obrázek 21: Diagram s přehledem funkcí dostupných pro administrátora

7.2.1 Číselníky

Archiv EFZ má dva druhy číselníků a dále nastavení firmy. První číselník obsahuje zákazníky, kterým se budou faktury rozesílat. Kromě standardních fakturačních a identifikačních údajů (jméno firmy, adresa, IC, DIC aj.) se každé firmě přiřazuje z druhého číselníku uživatel. Pokud by nebyl uživatel vyplněn, nemohly by se faktury distribuovat ven z archivu EFZ. Což však může být funkce v některých případech žádaná (např. zákazník nechce faktury dostávat z libovolných důvodů elektronicky, ale trvá na standardním papírové komunikaci) a faktury jsou zde pouze k archivaci. Je potřeba na tohle myslet při zadávání zákazníků do číselníku.

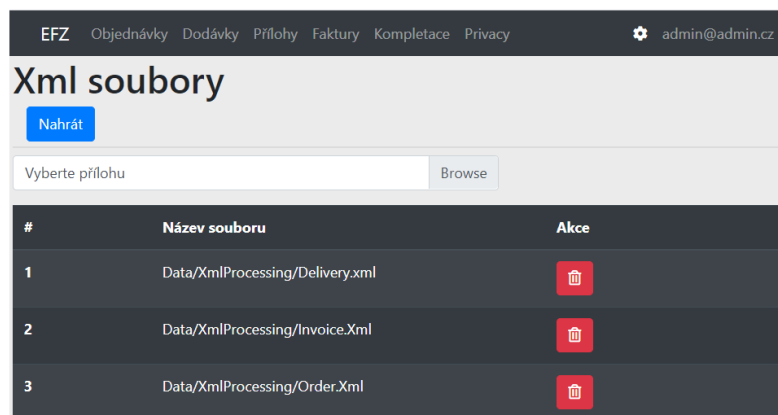
Druhým číselníkem jsou uživatelé, kterým se při zakládání nastavím pouze e-mail (ten slouží jako jméno uživatele pro přihlášení do systému), heslo a vyberu roli (tedy oprávnění, zda administrátor, nebo obyčejný uživatel). Poslední krok je výběr zákazníka. Opět, pokud nevyberu žádného zákazníka, bude účet aktivní, ale nebude mít přístup k žádným fakturám. Administrátor však uvidí doklady celého archivu EFZ bez omezení. Po založení nového uživatele jeho e-mailovým účtem, budou faktury odesílány na tento e-mail.

Poznámka k účtům, zákazník může mít registrováno více uživatelů, tedy přístup k dokladům jednoho zákazníka může být z více účtů, avšak jeden uživatel může vidět pouze jednoho zákazníka. Nejde nastavit, aby jeden zákazník viděl faktury od X zákazníků.

Správa firmy obsahuje identifikační údaje firmy (v našem případě správce portálu), které slouží jako podklad pro fakturaci → dotahují se do pole Dodavatel v hlavičce faktury.

7.2.2 Import dat

Archiv EFZ pracuje se dvěma druhy importu dat. XML import soubory v dohodnuté struktuře nahraje do lokálního úložiště, kde budou soubory čekat na zpracování automatickou úlohou. Pro XML import se nahrávají soubory tří různých datových struktur (objednávka, dodávka a faktura). Tlačítkem nahrát soubory nahrají na lokální úložiště, odkud je mohou v případě potřeby před zpracováním také smazat (obrázek č. 22).



Obrázek 22: XML vstupy do archivu EFZ

Druhý import funguje na stejném principu jako XML import, ale přílohy musí být ve formátu obrázkových souborů (*jpg*, *tiff* apod.). Stejně jako u XML lze soubory smazat, pokud je nechceme nechat zpracovat archivem.













Z důvodu prezentace funkčnosti archivu EFZ, je XML import, stejně jako import příloh v obrázkovém formátu nyní spouštěn manuálně. Administrátor musí ručně vybrat soubory z uložště dat (např. vstupní soubory typu **.xml* pro faktury a **.jpg* pro přílohy) a pomocí tlačítka je nahrát do archivu ke zpracování. Zautomatizovat celý proces importu je jen o nastavení cesty do uložště, odkud jednotlivé procesy data načítají.

7.2.3 Definice systémových úloh

Archiv EFZ má definované a nastavené tři úlohy, které vykonává v závislosti na nastavení. Parametrizace a vnitřní logika každé úlohy je nastavena v jádru systému, administrátor však může přes webové rozhraní nastavovat typy a časy spuštění každého procesu. Akce, které lze nastavit pro každou úlohou jsou následující:

- Cyklus úlohy: definice časového intervalu, ve kterém se bude úloha opakovaně spouštět (např. 10 minut po posledním spuštění)
- Spustit úlohu: přepne úlohu do automatického režimu
- Zastavit úlohu: vypne úlohu z automatického režimu
- Spustit manuálně: ruční okamžité zapnutí úlohy. Spustí se jednou, bez opakování

V přehledu systémových úloh se zobrazuje název úlohy, cyklus opakování, stav úlohy, datum a čas od posledního spuštění a výsledek běhu úlohy. Na konci přehledu jsou jednotlivé akce, které lze s úlohou provést (obrázek č. 23).

Joby						
#	Jméno	Cyklus jobu	Status	Poslední spuštění	poslední status	Akce
1	AttachmentScanProcessing	00:05:00	Zastaven	05/13/2020 13:32:16	V pořádku	   
2	XmlProcessing	00:05:00	Zastaven	05/13/2020 13:29:51	V pořádku	   
3	InvoiceCompletion	00:10:00	Zastaven	05/13/2020 13:35:44	V pořádku	   

Obrázek 23: Přehled systémových úloh




AttachmentScanProcessing – úloha vytěžování dat z čárových kódů nahraných příloh v souborech typu obrázek. V kapitole 7.2.2 jsem popsal funkci importu obrázkových dat. Tato data (tedy obrázky) musí být opatřeny čárovým kódem (aktuálně není jeho pozice v příloze definována, takže je jedno, kde se nachází), který si nese informaci o konkrétní dodávce. Úloha projde obrázek na vstupu, a pokud najde čárový kód, vyčte z něho informace. V případě úspěchu se obrázek konverzí převede do PDF dokumentu a uloží se do sekce Přílohy. Tam si obrázek, nyní již jako PDF dokument stáhnout. V případě nenalezení čárového kódu, nebo žádné shody se seznamem dodávek, skončí úloha chybou a příloha se nenahraje do archivu EFZ.

XmlProcessing – úloha vytěžování dat z XML souborů, které jsou uloženy v domluveném uložišti (viz kapitola 7.2.2). Úloha začne zpracovávat jednotlivé soubory a dle vnitřní struktury dat začíná nahrávat jednotlivé druhy dokladu (objednávka, dodávka, faktura) do jednotlivých modulů archivu EFZ. Po skončení úlohy budou jednotlivé dokumenty přístupné k náhledu v daných sekcích v podobě HTML formuláře. Pouze faktura se převede z HTML do PDF dokumentu a administrátor bude schopný si dokument z archivu EFZ stáhnout. Nejedná se však o platný doklad (neobsahuje elektronický podpis ani žádnou přílohu). Pokud během procesu zpracovávání dat najde úloha chybu, vypíše informaci do logu (viz. kapitola 7.2.4).

InvoiceCompletion – úloha kompletující faktury s přílohami, dle obchodního případu do jediného dokumentu, který opatří elektronickým podpisem a zajistí distribuci zákazníkům. Poté je doklad důvěryhodně uložen do archivu EFZ. Proces běží automaticky (popř. ručně), není závislý na žádném importu dat ani nečeká na kontroly ze strany uživatele. Jednou skončená úloha, která zkompletuje libovolný dokument, je neměnná a dokument již nelze nijak měnit. V případě potřeby změny, by se doklad musel vystavit znovu, opravit požadované změny a pak znovu zaslat do archivu EFZ ke zpracování. To jsou však organizační postupy každé organizační jednotky, které archiv EFZ nevyřeší.

7.2.4 Logy

Archiv EFZ obsahuje přímo v sekci plánovaných úloh zjednodušený logovací systém. Ten nemá nahradit plnotučná řešení (např. Cacti), ale má poskytnout jednoduché informace o tom, zda se jednotlivé úlohy dokončili úspěšně či nikoli (obrázek č. 24).

Joby					
#	Jméno	Datum jobu	Délka trvání	Status	Log
1	InvoiceCompletion	05/13/2020 13:35:44	00:00:05.0982535	V pořádku	
2	AttachmentScanProcessing	05/13/2020 12:19:31	00:00:06.0130998	Chyba	
3	XmlProcessing	05/13/2020 12:19:05	00:00:00.3755552	V pořádku	

Obrázek 24: Stav běhu spuštěných úloh

V kapitole 7.2.3 jsem definoval jednotlivé úlohy a jejich funkce. Pokud během procesu dojde k chybě, problém bude s největší pravděpodobností ve struktuře XML dat (dle logu zjistím pozici ve zdroji a případně chybu odstraním) nebo je chyba ve skenu přílohy (žádný nebo špatně čitelný kód, není shoda s žádným číslem dodávky, aj.). Chyby najdu v logu (obrázek č. 25)

Logy		×
Jméno	Chybové hlášení	
IMG_20200513_141541.jpg	Barcode not exist or not match to any delivery number.	

Obrázek 25: Výpis chyby neúspěšného importu naskenované přílohy

8 VYHODNOCENÍ PROJEKTU

Tato práce pojednává o tématu elektronické fakturace a snaží se tento segment mezisystémové komunikace mezi ekonomickými subjekty analyzovat z pohledu legislativního, tak i technického. Pohled na legislativu, jaká úskalí obnáší implementace elektronické fakturace do běžného života informačních systémů z pohledu zákona, jaké technologie se používají k naplnění shody se zákony a předpisy, o tom všem pojednává kapitola č. 3.

V kapitole č. 4 jsem se zaměřil na požadovaný archivační systém z pohledu obchodních procesů a požadavků, jak probíhá fakturace nyní a tedy jaké vstupy může nové řešení očekávat a jaké cíle na výstupu jsou od něho očekávány. Analýza je doplněna o prvek listinných příloh, které tvoří nedílnou součást požadavků na archivační systém a kterým musí vyhovět.

Technicky zaměřena kapitola č. 5 se zabývá technickým konceptem navrhnutého systému, pojmenovává finální produkt jako archiv EFZ (Elektronická fakturace zákazníkům) a definuje komunikační možnosti mezisystémové integrace. Metoda strukturovaných dat ve formátu XML je zde představena a zvolena pro praktickou část této práce.

Implementaci archivu EFZ se věnuje celá kapitola č. 6, která je svým rozsahem největší z celé práce, neboť musí představit a vysvětlit jednotlivé technologie použité při budování archivu EFZ, které výsledný produkt uvedou do provozu. Zabírá se použitými technologiemi (programovací jazyk, vývojové prostředí, použité knihovny a jejich implementace) a jejich praktickou implementací. Schematicky představím v této kapitole i vazby mezi moduly plánovače systémových procesů.

Nejpřínosnější je kapitola č. 7, ve které je v textové i grafické podobě představen archiv EFZ, který je postaven na poznatcích, analýzách a technologiích z kapitol 1 – 6. Archiv EFZ je představen jak z pohledu zákazníků (tedy uživatelů s omezením přístupu pouze ke svým fakturám, kdy jeden uživatel má přístup pouze k dokladům jednoho zákazníka), tak z pohledu administrátora celého řešení, který definuje procesy celého systému, spravuje číselníky (uživatelské i zákaznické) a v případě potřeby může v omezené míře zastoupit funkci importovaných dokladů, které jsou nutné ke kompletaci výsledných a důvěryhodných faktur.

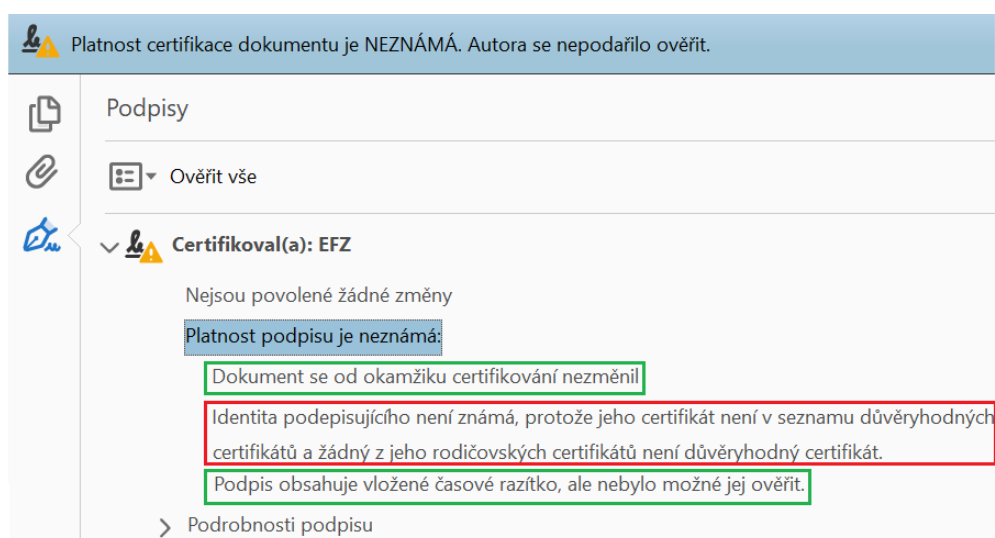
Kapitola č. 7 by měla představit a odprezentovat naplnění cílů této práce praktickou ukázkou, což se dle mě povedlo v plném rozsahu zadání. Na archiv EFZ byly kladeny nemalé požadavky, zejména pak z pohledu důvěryhodnosti a prokazatelnosti původu faktury, což společně s dalším požadavkem na kompletaci listinných příloh do podoby jediného dokumentu nebyly cíle malé. Po zprovoznění webového rozhraní archivu EFZ a prvních testech s importem a zpracováním dokladů se vyskytly potíže se zpracováním listinných příloh, respektive s jejich připojením do finálního dokumentu pomocí dat vytěžených s čárových kódů. Tohle bylo způsobeno zejména problémy s volně dostupnými knihovnami pro práci s čárovými kódy, které nebyly schopny data vytěžit. Pomohlo až integrování řešení Free Spire.Barcode, které v porovnání s ostatními zkoušenými produkty bylo funkční, nebylo omezené na definování konkrétních parametrů čárových kódů (např. pozice umístění čárového kódu na listinné příloze) a zvládlo data vytěžit. Samotný proces není nejrychlejší a v závislosti na velikosti vstupního obrazového souboru čtení trvá, ale práci odvede dobře. V komerční verzi je tento problém vyřešen, na potřeby této práce však bezplatná verze stačila.

Testována byla také varianta s vložením více čárových kódů na jednu listinnou přílohu a i s tímhle si archiv EFZ při zpracování poradil bez problému. Náhled na takovou zpracovanou přílohu je na obrázku č. 26.



Obrázek 26: Úspěšně zpracování přílohy s více čárovými kódy

Dalším větším problém k vyřešení bylo podepisování dokumentů elektronickým podpis a zároveň vložení časového razítka nezávislou časovou autoritou. Je mnoho dostupných řešení, které po implementaci dokáží pracovat s elektronickým podpisem (více v kapitole 6.6), problém však nastává ve chvíli, kdy se vlastnoručně vygenerovaným certifikátem snažím získat časový token ze serverů nezávislých časových autorit. I když jsou některé servery volně dostupné a nabízí funkčnost časových razítek, tak jakmile je dokument podepsaný vlastnoručním certifikátem místo kvalifikovaného certifikátu pro elektronický podpis vydaný např. první certifikační autoritou (nebo jinou akreditovanou institucí, např. Českou poštou) bude výsledný dokument hlásit problém s ověřením certifikátu. Problém je zobrazen na obrázku č. 27.



Obrázek 27: Problém s ověřením certifikátu

Obrázek č. však také zobrazuje splnění požadavků na výsledný dokument, tedy dodržení neměnitelnosti dokumentu od data jeho vytvoření (více v kapitole 3.1) a také prokázání že v této podobě existovaly před časem uvedeným v časovém razítku. Podrobnosti certifikátu si lze zobrazit v panelu dokumentu, pokud se vybere položka Podrobnosti certifikátu, uvidím detaily vygenerovaného certifikátu použitého pro podepisování dokumentů v archivu EFZ.

Elektronická příloha diplomové práce bude obsahovat jednu kompletní fakturu, která dokazuje splnění vytyčených cílů, tedy propojení faktury vystavené v extérním ERP, doplněnou o naskenovanou přílohu, se kterou byla propojena na základě vytěžení dat z čárového kódu, a celý tento dokument se elektronicky podepsal certifikátem, obdržel časové razítko a nakonec byl distribuován zákazníkovi.

Než ukončím práci závěrečnou kapitolou, stručně zmíním možný směr a doplnění funkčnosti archivu EFZ, kdyby se pokračovalo ve vývoji. Nyní archiv splňuje minimální základ funkčnosti, nutný pro naplnění legislativních a businessových požadavků. Není zde však nastavena pro toto minimum plná automatika, tedy ve smyslu generování náhodných dat v dohodnuté struktuře na domluvené uložení, odkud úlohy archivu EFZ načítají vstupní data. Prezentováno bylo vše manuálním režimem, avšak systém je na automatiku připraven. Jen jsem nebyl schopný v čase potřebném na tuto práci nastavit právě tu generickou činnost pro vstupní data, se kterou by byl archiv EFZ zajímavější.

Stejně tak není nyní implementován žádný systém, který by hlídal stavy faktur. Není to nutné pro funkčnost a shodu s legislativou, ale opět z pohledu business je to pravděpodobně nutný požadavek, kdy dodavatel (tedy správce archivu EFZ a zároveň vystavovatel faktur) potřebuje evidovat stavy u faktur, zda si je zákazník převzal z archivu EFZ, nebo ne, zda byla odeslána notifikace o nové faktuře apod. Velice jednoduchý reporting, který by hlídal právě předem domluvené stavy, by měl efektivní přínos pro archiv EFZ.

Stejně tak by se daly nastavit typy faktur, ve smyslu dohody se zákazníky. Ne všichni by chtěli přejít na elektronickou fakturaci hned, ale řekněme někdy v budoucnu. V takovém případě by neměli mít přístup k archivovaným fakturám, které již portál EFZ archivuje. Viděli by pouze dokumenty od data registrace. Popř. jsou faktury, které v době vystavené nemají ještě přílohy. Proces by však faktury zkompletoval bez nich a uzavřel pro další změny. Bylo by potřeba ošetřit i tuto variantu. Nejen tato praktická řešení by se dala zahrnout do příznaku na typech faktur v podobě rozšířených metadat. Tohle je však diskuze na budoucí případně využívání a rozšiřování archivu EFZ.

ZÁVĚR

Cílem této diplomové práce bylo vytvořit fakturační portál pro důvěryhodnou archivaci faktur v elektronické podobě. Vstupy do archivu by přicházely z externích ERP systémů a zároveň by k archivovaným fakturám dokázal připojit jejich listinné přílohy na základě dat, které si každá příloha nese uvnitř čárového kódu v každém dokumentu. Faktury zkompletované s přílohami bude portál archivovat v důvěryhodné podobě a zároveň je distribuovat zákazníkům.

Na základě zadání jsem provedl analýzu potřebnou pro vytvoření jednoduché webové aplikace, která dokáže zpracovat všechny vstupy (faktury z ERP, přílohy získané z multifunkčních nebo mobilních zařízení) a dle potřeby na výstupu dodat kompletní dokument, který je v čase neměnný a zároveň je ve shodě s legislativními požadavky České republiky a Evropskou směrnicí pro formáty a syntaxe faktur. Takový dokument je připraven k vyzvednutí přes webové rozhraní libovolného internetového prohlížeče, z mobilního zařízení nebo je distribuován ven z portálu zákazníkům.

V průběhu implementace jsem narazil na nemalé problémy (zejména s vytěžením dat z čárového kódu a také při získávání časových razítek při podepisování dokumentů nekvalifikovaným komerčním certifikátem), ale nakonec se podařilo cílové portálové řešení spustit se všemi funkcemi definovanými v zadání a splnit tak všechny cíle práce.

Výsledný archiv EFZ je plně funkční a i s minimem funkcí by se dal s menšími úpravami nasadit do produkce. Je tam velký prostor pro budoucí vývoj a rozvoj portálu, ale věřím, že práce na podobném řešení má smysl a pro uživatele může být portál velice přínosný (nejen z pohledu efektivity procesů, ale také z pohledu snižování nákladů na papírovou komunikaci a urychlení toku dat mezi ekonomickými subjekty).

LITERATURA

- [1] Legislativa (Legislation). In: ManagementMania.com [online]. Wilmington (DE) 2011-2020, 30.07.2017 [cit. 2020-05-04]. Dostupné z: <https://managementmania.com/cs/legislativa>
- [2] Elektronická fakturace. Ministerstvo financí ČR [online]. Praha: Ministerstvo financí, 2018 [cit. 2020-05-10]. Dostupné z: <https://www.mfcr.cz/cs/verejny-sektor/elektronicka-fakturace/zakladni-informace>
- [3] Elektronická časová razítka. První certifikační autorita, a.s. (I.CA) [online]. Praha: První certifikační autorita, a.s. (I.CA) [cit. 2020-05-10]. Dostupné z: <https://www.ica.cz/elektronicka-casova-razitka>
- [4] FlexiBee XML [online]. Praha, 2014 [cit. 2020-05-09]. Dostupné z: <https://www.flexibee.eu/api/dokumentace/ref/xml/>
- [5] ŠULC, Přemysl. Vývoj webových aplikací na platformě ASP.NET MVC a Single age Application [online]. Hradec Králové, 2017 [cit. 2020-05-13]. Dostupné z: <https://theses.cz/id/tq859n/>. Diplomová práce. Univerzita Hradec Králové.
- [6] DinkToPdf. *GitHub* [online]. 2017 [cit. 2020-05-09]. Dostupné z: <https://github.com/rdvojmoc/DinkToPdf>
- [7] IText 7 .NET (C#). IText PDF [online]. [cit. 2020-05-09]. Dostupné z: <https://itextpdf.com/en/resources/api-documentation/itext-7-net>
- [8] Tutorials Spire.BarCode. E - iceblue [online]. [cit. 2020-05-09]. Dostupné z: <https://www.e-iceblue.com/Tutorials/Spire.BarCode.html>
- [9] PDFsharp Samples. PDFsharp and MigraDoc [online]. Germany: empira Software [cit. 2020-05-11]. Dostupné z: <http://pdfsharp.net/wiki/PDFsharpSamples.ashx>
- [10] KÜHNHAUSER, Karl-Heinz. ABAP: výukový kurz. Brno: Computer Press, 2009. ISBN 978-802-5121-177.
- [11] MAASSEN, André. SAP R/3: kompletní průvodce. Brno: Computer Press, 2007. Informační systémy. ISBN 978-802-5117-507.
- [12] GAMMA, Erich. Návrh programů pomocí vzorů: stavební kameny objektově orientovaných programů. Praha: Grada, 2003. Moderní programování. ISBN 80-247-0302-5

PŘÍLOHY

Přílohy nahrané do IS EDISON obsahují tato data:

- Diplomová práce ve formátu pdf/A
- Aplikaci v archivu „EFZ.zip“ – komprimace v zip souboru
- Schéma plánovače úloh (JobSchedulerBIProvider.xps)